

# A Unifying Approach to Efficient (Near)-Gathering of Disoriented Robots with Limited Visibility

Jannik Castenow ✉ 

Heinz Nixdorf Institute & Computer Science Department, Paderborn University, Germany

Jonas Harbig ✉ 

Heinz Nixdorf Institute & Computer Science Department, Paderborn University, Germany

Daniel Jung ✉ 

Heinz Nixdorf Institute & Computer Science Department, Paderborn University, Germany

Peter Kling ✉ 

Department of Informatics, Universität Hamburg, Germany

Till Knollmann ✉ 

Heinz Nixdorf Institute & Computer Science Department, Paderborn University, Germany

Friedhelm Meyer auf der Heide ✉

Heinz Nixdorf Institute & Computer Science Department, Paderborn University, Germany

---

## Abstract

We consider a swarm of  $n$  robots in a  $d$ -dimensional Euclidean space. The robots are oblivious (no persistent memory), disoriented (no common coordinate system/compass), and have limited visibility (observe other robots up to a constant distance). The basic formation task GATHERING requires that all robots reach the same, not predefined position. In the related NEAR-GATHERING task, they must reach distinct positions in close proximity such that every robot sees the entire swarm. In the considered setting, GATHERING can be solved in  $\mathcal{O}(n + \Delta^2)$  synchronous rounds both in two and three dimensions, where  $\Delta$  denotes the initial maximal distance of two robots [3, 13, 24].

In this work, we formalize a key property of efficient GATHERING protocols and use it to define  $\lambda$ -contracting protocols. Any such protocol gathers  $n$  robots in the  $d$ -dimensional space in  $\mathcal{O}(\Delta^2)$  synchronous rounds, for  $d \geq 2$ . For  $d = 1$ , any  $\lambda$ -contracting protocol gathers in optimal time  $\mathcal{O}(\Delta)$ . Moreover, we prove a corresponding lower bound stating that any protocol in which robots move to target points inside the local convex hulls of their neighborhoods –  $\lambda$ -contracting protocols have this property – requires  $\Omega(\Delta^2)$  rounds to gather all robots ( $d > 1$ ). Among others, we prove that the  $d$ -dimensional generalization of the GTC-protocol [3] is  $\lambda$ -contracting. Remarkably, our improved and generalized runtime bound is independent of  $n$  and  $d$ .

We also introduce an approach to make any  $\lambda$ -contracting protocol collision-free (robots never occupy the same position) to solve NEAR-GATHERING. The resulting protocols maintain the runtime of  $\Theta(\Delta^2)$  and work even in the semi-synchronous model. This yields the first NEAR-GATHERING protocols for disoriented robots and the first proven runtime bound. In particular, combined with results from [28] for robots with global visibility, we obtain the first protocol to solve UNIFORM CIRCLE FORMATION (arrange the robots on the vertices of a regular  $n$ -gon) for oblivious, disoriented robots with limited visibility.

**2012 ACM Subject Classification** Theory of computation → Distributed algorithms

**Keywords and phrases** mobile robots, gathering, limited visibility, runtime, collision avoidance, near-gathering

**Digital Object Identifier** 10.4230/LIPIcs.OPODIS.2022.15

**Related Version** Full Version: <https://arxiv.org/abs/2206.07567> [14]

**Funding** This work was partially supported by the German Research Foundation (DFG) under the project number ME 872/14-1.



© Jannik Castenow, Jonas Harbig, Daniel Jung, Peter Kling, Till Knollmann, and Friedhelm Meyer auf der Heide;

licensed under Creative Commons License CC-BY 4.0

26th International Conference on Principles of Distributed Systems (OPODIS 2022).

Editors: Eshcar Hillel, Roberto Palmieri, and Etienne Rivière; Article No. 15; pp. 15:1–15:25

Leibniz International Proceedings in Informatics



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

## 1 Introduction

Envision a huge swarm of  $n$  robots spread in a  $d$ -dimensional Euclidean space that must solve a *formation task* like GATHERING (moving all robots to a single, not pre-determined point) or UNIFORM-CIRCLE (distributing the robots over the vertices of a regular  $n$ -gon). Whether and how efficiently a given task is solvable varies largely with the robots' capabilities (local vs. global visibility, memory vs. memory-less, communication capabilities, common orientation vs. disorientation). While classical results study what capabilities the robots need *at least* to solve a given task, our focus lies on *how fast* a given formation task can be solved assuming simple robots. Specifically, we consider the GATHERING problem and the related NEAR-GATHERING problem for oblivious, disoriented robots with a limited viewing range.

GATHERING is the most basic formation task and a standard benchmark to compare robot models [27]. The robots must gather at the same, not predefined, position. Whether or not GATHERING is solvable depends on various robot capabilities. It is easy to see that robots can solve GATHERING in case they have unlimited visibility (can observe all other robots) and operate fully synchronously [17]. However, as soon as the robots operate asynchronously, have only limited visibility, or do not agree on common coordinate systems, the problem gets much harder or even impossible to solve (see Section 1.1 for a comprehensive discussion). A well-known protocol to solve GATHERING of robots with limited visibility is the GO-TO-THE-CENTER (GTC) protocol that moves each robot towards the center of the smallest enclosing circle of all observable robots [3]. GTC gathers all robots in  $\mathcal{O}(n + \Delta^2)$  synchronous rounds, where the *diameter*  $\Delta$  denotes the initial maximal distance of two robots [24]. The term  $n$  upper bounds the number of rounds in which robots *collide* (move to the same position), while  $\Delta^2$  results from how quickly the global smallest enclosing circle shrinks. Hence, GTC not only forces the robots to collide in the final configuration but also incurs several collisions during GATHERING. Such collisions are fine for point robots in theoretical models but a serious problem for physical robots that cannot occupy the same position. This leads us to NEAR-GATHERING, which requires the robots to move *collision-free* to *distinct* locations such that every robot can observe the entire swarm despite its limited visibility [42]. Requiring additionally that, eventually, robots *simultaneously* (within one round/epoch) *terminate*, turns NEAR-GATHERING into a powerful subroutine for more complex formation tasks like UNIFORM CIRCLE. Once all robots see the entire swarm and are simultaneously aware of that, they can switch to the protocol of [28] to build a uniform circle. Although that protocol is designed for robots with a global view, we can use it here since solving NEAR-GATHERING grants the robots *de facto* a global view. Note the importance of *simultaneous* termination, as otherwise, some robots might build the new formation while others are still gathering, possibly disconnecting some robots from the swarm.

**Robot Model.** We assume the standard *OBLOT* model [27] for oblivious, point-shaped robots in  $\mathbb{R}^d$ . The robots are *anonymous* (no identifiers), *homogeneous* (all robots execute the same protocol), *identical* (same appearance), *autonomous* (no central control) and *deterministic*. Moreover, we consider *disoriented* robots with *limited visibility*. Disorientation means that a robot observes itself at the origin of its local coordinate system, which can be arbitrarily rotated and inverted compared to other robots. The disorientation is *variable*, i.e., the local coordinate system might differ from round to round. Limited visibility implies that each robot can observe other robots only up to a constant distance. The robots do not have *multiplicity detection*, i.e., robots observe only a single robot in case multiple robots are located at the same position. Furthermore, time is divided into discrete LCM-cycles (*rounds*)

consisting of the operations LOOK, COMPUTE and MOVE. During its LOOK operation, a robot takes a snapshot of all visible robots, which is used in the following COMPUTE operation to compute a *target point*, to which the robot moves in the MOVE operation. Moves are *rigid* (a robot always reaches its target point) and depend solely on observations from the last LOOK operation (robots are oblivious). The time model can be fully synchronous ( $\mathcal{F}\text{SYNC}$ ; all robots are active each round and operations are executed synchronously), semi-synchronous ( $\mathcal{S}\text{SYNC}$ ; a subset of robots is active each round and operations are executed synchronously), or completely asynchronous ( $\mathcal{A}\text{SYNC}$ ). The  $\mathcal{S}\text{SYNC}$  and  $\mathcal{A}\text{SYNC}$  schedules of the robots are *fair*, i.e., each robot is activated infinitely often. Time is measured in rounds in  $\mathcal{F}\text{SYNC}$  and *epochs* (the smallest number of rounds such that all robots finish one LCM-cycle) in  $\mathcal{S}\text{SYNC}$  or  $\mathcal{A}\text{SYNC}$ .

**Results in a Nutshell.** For GATHERING of oblivious, disoriented robots with limited visibility in  $\mathbb{R}^d$ , we introduce the class of  $\lambda$ -contracting protocols for a constant  $\lambda \in (0, 1]$ . For instance, the well-known GTC [3] and several other GATHERING protocols are  $\lambda$ -contracting. We prove that for  $d > 1$ , every  $\lambda$ -contracting protocol gathers a swarm of diameter  $\Delta$  in  $\mathcal{O}(\Delta^2)$  rounds. The case  $d = 1$  leads to an optimal time bound of  $\Theta(\Delta)$  rounds. We also prove a matching lower bound for any protocol in which robots always move to points inside the convex hull of their neighbors, including themselves. While our results for GATHERING assume the  $\mathcal{F}\text{SYNC}$  scheduler<sup>1</sup>, for NEAR-GATHERING we also consider  $\mathcal{S}\text{SYNC}$ . We show how to transform any  $\lambda$ -contracting protocol into a *collision-free*  $\lambda$ -contracting protocol to solve NEAR-GATHERING while maintaining a runtime of  $\mathcal{O}(\Delta^2)$ .

## 1.1 Related Work

One important topic of the research area of distributed computing by mobile robots is *pattern formation* problems, i.e., the question of which patterns can be formed by a swarm of robots and which capabilities are required. For instance, the ARBITRARY PATTERN FORMATION problem requires the robots to form an arbitrary pattern specified in the input [21, 25, 30, 46, 47, 48]. The patterns *point* and *uniform circle* play an important role since these are the only two patterns that can be formed starting from *any* input configuration due to their high symmetry [46]. In the following, we focus on the pattern *point*, more precisely on the GATHERING, CONVERGENCE and NEAR-GATHERING problems. While GATHERING requires that all robots move to a single (not predefined) point in finite time, CONVERGENCE demands that for all  $\varepsilon > 0$ , there is a point in time such that the maximum distance of any pair of robots is at most  $\varepsilon$  and this property is maintained (the robots *converge* to a single point). NEAR-GATHERING is closely related to the CONVERGENCE problem by robots with limited visibility. Instead of converging to a single point, NEAR-GATHERING is solved as soon as all robots are located at *distinct* locations within a small area. For a more comprehensive overview of other patterns and models, we refer to [26].

**Possibilities & Impossibilities.** In the context of robots with *unlimited visibility*, GATHERING can be solved under the  $\mathcal{F}\text{SYNC}$  scheduler by disoriented and oblivious robots without multiplicity detection [17]. Under the same assumptions, GATHERING is impossible under the  $\mathcal{S}\text{SYNC}$  and  $\mathcal{A}\text{SYNC}$  schedulers [45]. Multiplicity detection plays a crucial role: at least 3 disoriented robots with multiplicity detection can be gathered in  $\mathcal{A}\text{SYNC}$  (and thus also

<sup>1</sup> With the considered robot capabilities, GATHERING is impossible in  $\mathcal{S}\text{SYNC}$  or  $\mathcal{A}\text{SYNC}$  [45].

$\mathcal{S}\text{SYNC}$ ) [16]. The case of 2 robots remains impossible [46]. Besides multiplicity detection, an agreement on one axis of the local coordinate systems also allows the robots to solve GATHERING in  $\mathcal{A}\text{SYNC}$  [5]. CONVERGENCE requires less assumptions than GATHERING. No multiplicity detection is needed for the  $\mathcal{A}\text{SYNC}$  scheduler [17].

Under the assumption of *limited visibility*, disoriented robots without multiplicity detection can be gathered in  $\mathcal{F}\text{SYNC}$  [3] with the GTC protocol that moves every robot towards the center of the smallest circle enclosing its neighborhood. GTC has also been generalized to three dimensions [13]. In  $\mathcal{A}\text{SYNC}$ , current solutions require more capabilities: GATHERING can be achieved by robots with limited visibility that agree additionally on the axes and orientation of their local coordinate systems [29]. It is open whether fewer assumptions are sufficient to solve GATHERING of robots with limited visibility in  $\mathcal{S}\text{SYNC}$  or  $\mathcal{A}\text{SYNC}$ . In  $\mathcal{S}\text{SYNC}$ , CONVERGENCE can be solved even by disoriented robots with limited visibility without multiplicity detection [3]. However, similar to GATHERING, it is still open whether disoriented robots with limited visibility can solve CONVERGENCE under the  $\mathcal{A}\text{SYNC}$  scheduler. Recently, it could be shown that multiplicity detection suffices to solve CONVERGENCE under the more restricted  $k$ - $\mathcal{A}\text{SYNC}$  scheduler. The constant  $k$  bounds how often other robots can be activated within one LCM cycle of a single robot [36, 37].

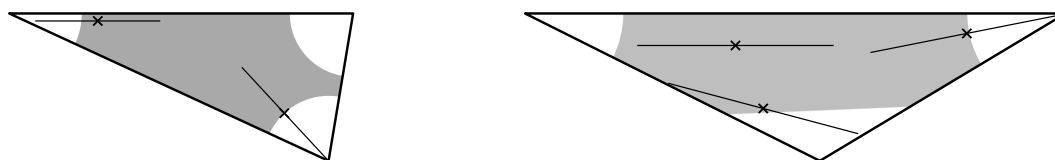
The NEAR-GATHERING problem has been introduced in [41, 42] together with a protocol to solve NEAR-GATHERING by robots with limited visibility and agreement on one axis of their local coordinate systems under the  $\mathcal{A}\text{SYNC}$  scheduler. An important tool to prevent collisions is a well-connected initial configuration, i.e., the initial configuration is connected concerning the *connectivity range* which is by an additive constant smaller than the viewing range [41, 42]. In earlier work, NEAR-GATHERING has been used as a subroutine to solve ARBITRARY PATTERN FORMATION by robots with limited visibility [49]. The solution, however, uses infinite persistent memory for each robot. Further research directions study GATHERING and CONVERGENCE under crash faults or Byzantine faults [2, 4, 6, 7, 8, 9, 10, 11, 22, 32, 43] or inaccurate measurement and movement sensors of the robots [18, 31, 33, 37].

**Runtime.** Considering disoriented robots with *unlimited* visibility, it is known that CONVERGENCE can be solved in  $\mathcal{O}(n \cdot \log \Delta/\varepsilon)$  epochs under the  $\mathcal{A}\text{SYNC}$  scheduler, where the diameter  $\Delta$  denotes the initial maximum distance of two robots [19] (initially a bound of  $\mathcal{O}(n^2 \cdot \log \Delta/\varepsilon)$  has been proven in [17]). When considering disoriented robots with *limited* visibility and the  $\mathcal{F}\text{SYNC}$  scheduler, the GTC protocol solves GATHERING both in two and three dimensions in  $\Theta(n + \Delta^2)$  rounds [13, 24]. It is conjectured that the runtime is optimal in worst-case instances, where  $\Delta \in \Omega(n)$  [13, 15]. There is some work achieving a faster runtime for slightly different models: robots on a grid in combination with the  $\mathcal{LUMI}$  model (constant sized local communication via lights) [1, 20], predefined neighborhoods in a closed chain [1, 15] or agreement on one axis of the local coordinate systems [44]. Also, a different time model – the *continuous* time model, where the movement of robots is defined for each *real* point in time by a bounded velocity vector – leads to a faster runtime: There are protocols with a runtime of  $\mathcal{O}(n)$  [12, 23]. In [38], a more general class of continuous protocols has been introduced, the *contracting* protocols. Contracting protocols demand that each robot part of the global convex hull of all robots' positions moves at full speed towards the inside. Any contracting protocol gathers all robots in time  $\mathcal{O}(n \cdot \Delta)$ . One such protocol also needs a runtime of  $\Omega(n \cdot \Delta)$  in a specific configuration. For instance, the continuous variant of GTC is contracting [38] but also the protocols of [12, 23]. The class of contracting protocols also generalizes to three dimensions with an upper time bound of  $\mathcal{O}(n^{3/2} \cdot \Delta)$  [13].

## 1.2 Our Contribution & Outline

In the following, we provide a detailed discussion of our results and put them into context concerning the related results discussed in Section 1.1. Our results assume robots located in  $\mathbb{R}^d$  and the *OBLLOT* model for deterministic, disoriented robots with limited visibility.

**Gathering.** Our first main contribution is introducing a large class of *GATHERING* protocols in  $\mathcal{F}_{\text{SYNC}}$  that contains several natural protocols such as *GTC*. We prove that *every* protocol from this class gathers in  $\mathcal{O}(\Delta^2)$  ( $d > 1$ ) or  $\mathcal{O}(\Delta)$  ( $d = 1$ ) rounds, where the diameter  $\Delta$  denotes the initial maximal distance between two robots. Note that, the bound of  $\mathcal{O}(\Delta^2)$  not only reflects how far a given initial swarm is from a gathering but also improves the *GTC* bound from  $\mathcal{O}(n + \Delta^2)$  to  $\mathcal{O}(\Delta^2)$ . We call this class  $\lambda$ -contracting protocols. Such protocols restrict the allowed target points to a specific subset of a robot’s local convex hull (formed by the positions of all visible robots, including itself) in the following way. Let  $diam$  denote the diameter of a robot’s local convex hull. Then, a target point  $p$  is an allowed target point if it is the center of a line segment of length  $\lambda \cdot diam$ , completely contained in the local convex hull. This guarantees that the target point lies far enough inside the local convex hull (at least along one dimension) to decrease the swarm’s diameter sufficiently. See Figure 1 for an illustration.



■ **Figure 1** Two local convex hulls, each formed by 3 robots. The gray area marks valid target points of  $\lambda$ -contracting protocols. The exemplary line segments all have length  $\lambda \cdot diam$ , where  $diam$  is the diameter of the respecting convex hull. On the left  $\lambda = 4/7$ , on the right  $\lambda = 4/11$ .

We believe these  $\lambda$ -contracting protocols encapsulate the core property of fast *GATHERING* protocols. Their analysis is comparatively clean, simple, and holds for any dimension  $d$ . Thus, by proving that (the generalization of) *GTC* is  $\lambda$ -contracting for arbitrary dimensions, we give the first protocol that provably gathers in  $\mathcal{O}(\Delta^2)$  rounds for any dimension. As a strong indicator that our protocol class might be asymptotically optimal, we prove that every *GATHERING* protocol for deterministic, disoriented robots whose target points lie *always inside* the robots’ local convex hulls requires  $\Omega(\Delta^2)$  rounds. Staying in the convex hull of visible robots is a natural property for any known protocol designed for oblivious, disoriented robots with limited visibility. Thus, reaching a sub-quadratic runtime – if at all possible – would require the robots to compute target points outside of their local convex hulls sufficiently often.

**Near-Gathering.** Our second main contribution proves that any  $\lambda$ -contracting protocol for *GATHERING* can be transformed into a collision-free protocol that solves *NEAR-GATHERING* in  $\mathcal{O}(\Delta^2)$  rounds ( $\mathcal{F}_{\text{SYNC}}$ ) or epochs ( $\mathcal{S}_{\text{SYNC}}$ ). As in previous work [41, 42], our transformed protocols require that the initial swarm is *well-connected*, i.e., the swarm is connected concerning the *connectivity range* of 1 and the robots have a viewing range of  $1 + \tau$ , for a constant  $\tau > 0$ . The adapted protocols ensure that the swarm stays connected concerning the connectivity range.

The well-connectedness serves two purposes. First, it allows a robot to compute its target point under the given  $\lambda$ -contracting protocol and the target points of nearby robots to prevent collisions. Its second purpose is to enable termination: Once there is a robot whose local convex hull has a diameter at most  $\tau$ , *all* robots must have distance at most  $\tau$ , as otherwise, the swarm would not be connected concerning the connectivity range 1. Thus, all robots can *simultaneously* decide (in the same round in  $\mathcal{F}\text{SYNC}$  and within one epoch in  $\mathcal{S}\text{SYNC}$ ) whether NEAR-GATHERING is solved. If the swarm is not well-connected, it is easy to see that such a simultaneous decision is impossible<sup>2</sup>. The simultaneous termination also allows us to derive the first protocol to solve UNIFORM-CIRCLE for disoriented robots with limited visibility. Once the robots' local diameter (and hence also the global diameter) is less than  $\tau$ , they essentially have a global view. As the UNIFORM CIRCLE protocol from [28] maintains a small diameter, it can be used after the termination of our NEAR-GATHERING protocol without any modification.

**Outline.** Section 2 introduces various notations.  $\lambda$ -contracting protocols are introduced in Section 3.1. Upper and lower runtime bounds are provided in Section 3.2. The section is concluded with three exemplary  $\lambda$ -contracting protocols, including GTC (Section 3.3). Section 4 discusses the general approach to transform any  $\lambda$ -contracting protocol (in any dimension) into a collision-free protocol to solve NEAR-GATHERING. Finally, the paper is concluded, and future research questions are addressed in Section 5. Due to space constraints, some proofs and additional information are moved to the full version of this paper [14].

## 2 Notation

We consider a swarm of  $n$  robots  $R = \{r_1, \dots, r_n\}$  moving in a  $d$ -dimensional Euclidean space  $\mathbb{R}^d$ . Initially, the robots are located at pairwise distinct locations. We denote by  $p_i(t)$  the position of robot  $r_i$  in a global coordinate system (not known to the robots) in round  $t$ . Robots have a *limited visibility*, i.e., they can observe other robots only up to a constant distance. We distinguish the terms *viewing* range and *connectivity* range and normalize all distances such that the connectivity range is 1. The initial configuration is connected concerning the connectivity range. More formally,  $\text{UBG}(t) = (R, E(t))$  the Unit Ball Graph, where  $\{r_i, r_j\} \in E(t)$  if and only if  $|p_i(t) - p_j(t)| \leq 1$ , where  $|\cdot|$  represents the Euclidean norm. The initial Unit Ball Graph  $\text{UBG}(0)$  is always connected. The connectivity and viewing ranges are equal when we study the GATHERING problem. In the context of NEAR-GATHERING, the viewing range is larger than the connectivity range. More formally, the viewing range is  $1 + \tau$ , for a constant  $0 < \tau \leq 2/3$ . Thus, the robots can observe other robots at a distance of at most  $1 + \tau$ . Two robots are neighbors at round  $t$  if their distance is at most the viewing range (1 for GATHERING and  $1 + \tau$  for NEAR-GATHERING). Due to their viewing range, all robots have a common understanding of 1 and  $1 + \tau$  (1 and  $\tau$  are known to the robots). The set  $N_i(t)$  contains all neighbors of  $r_i$  in round  $t$ , including  $r_i$ . Additionally,  $\text{hull}_i^t$  denotes the *local convex hull* of all neighbors of  $r_i$ , i.e., the smallest convex polytope that encloses the positions of all robots in  $N_i(t)$ , including  $r_i$ . We define  $\text{diam}(t)$  as the maximum distance of any pair of robots at time  $t$ . Moreover,  $\Delta := \text{diam}(0)$ , i.e., the maximum distance of any pair of robots in the initial configuration. Lastly,  $\text{diam}_i(t)$  denotes the maximum distance of any two neighbors of  $r_i$  in round  $t$ .

<sup>2</sup> Consider a protocol that solves NEAR-GATHERING for a swarm of two robots and terminates in the  $\mathcal{F}\text{SYNC}$  model. Fix the last round before termination and add a new robot visible to only one robot (the resulting swarm is not connected concerning). One of the original two robots still sees the same situation as before and will terminate, although NEAR-GATHERING is not solved.

**Discrete Protocols.** A discrete robot formation protocol  $\mathcal{P}$  specifies for every round  $t \in \mathbb{N}_0$  how each robot determines its target point, i.e., it is an algorithm that computes the target point  $\text{target}_i^{\mathcal{P}}(t)$  of each robot in the COMPUTE operation based upon its snapshot taken during LOOK. To simplify the notation,  $\text{target}_i^{\mathcal{P}}(t)$  might express the target point of  $r_i$  either in the local coordinate system of  $r_i$  or in a global coordinate system (not known to  $r_i$ ) – the concrete meaning is always clear based on the context. Finally, during MOVE, each robot moves to the position computed by  $\mathcal{P}$ , i.e.,  $p_i(t+1) = \text{target}_i^{\mathcal{P}}(t)$  for all robots  $r_i$ .

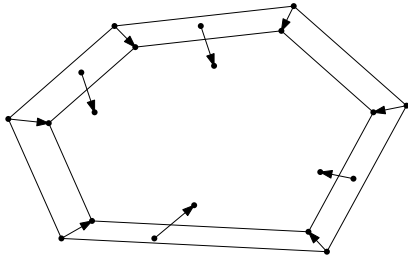
**Problem Statements.** GATHERING requires all robots to gather at a single, not predefined point. While the GATHERING problem clearly demands that more than one robot occupies the same position, this is prohibited in the NEAR-GATHERING problem. Two robots  $r_i$  and  $r_j$  *collide* in round  $t$  if  $p_i(t) = p_j(t)$ . A discrete robot formation protocol is *collision-free*, if there is no round  $t' \in \mathbb{N}_0$  with a collision. NEAR-GATHERING requires all robots to maintain distinct locations, become mutually visible, and be aware of this fact in the same round/epoch. More formally, NEAR-GATHERING is solved if there is a time  $t' \in \mathbb{N}_0$  and a constant  $0 \leq c_{ng} \leq 1$  such that  $\text{diam}(t') \leq c_{ng}$ ,  $p_i(t'') = p_i(t')$  for all robots  $r_i$  and all rounds  $t'' \geq t'$  and  $p_i(t) \neq p_j(t)$  for all robots  $r_i$  and  $r_j$  and rounds  $t$ . Moreover, all robots terminate simultaneously, i.e., know in the same round or within one epoch that  $\text{diam}(t) < c_{ng}$ . In our protocols,  $c_{ng} = \tau$ . Our protocols keep  $\text{UBG}(t)$  always connected and hence, robots can detect termination as soon as  $\text{diam}(t) < \tau$  (due to their viewing range of  $1 + \tau$ ). Due to the disorientation and obliviousness of the robots, any protocol to solve NEAR-GATHERING must be collision-free. As soon as two robots would move to the same location, their neighborhoods are identical and their local coordinate systems could have the same orientation such that the two robots would always compute the same movement. Hence, the robots cannot deterministically move to different locations. As a consequence, collisions must be avoided.

### 3 A Class of Gathering Protocols

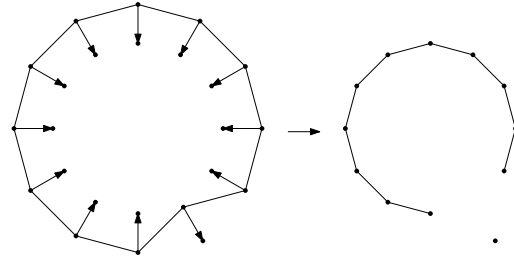
In this section, we describe the class of  $\lambda$ -contracting (gathering) protocols – a class of protocols which solve GATHERING in  $\Theta(\Delta^2)$  rounds and serves as a basis for collision-free protocols to solve NEAR-GATHERING (see Section 4). Moreover, we derive a subclass of  $\lambda$ -contracting protocols, called  $(\alpha, \beta)$ -contracting protocols. The class of  $(\alpha, \beta)$ -contracting protocols is a powerful tool to determine whether a given gathering protocol (such as GTC) fulfills the property of being  $\lambda$ -contracting.

The first intuition to define a class of protocols to solve GATHERING would be to transfer the class of continuous contracting protocols (cf. Section 1.1) to the discrete LCM case. A continuous robot formation protocol is called *contracting* if robots that are part of the global convex hull move with constant speed towards the inside or along the boundary of the global convex hull. A translation to the discrete (LCM) case might be to demand that each robot moves a constant distance inwards (away from the boundary) of the global convex hull, cf. Figure 2.

However, such a protocol cannot exist in the discrete LCM setting. Consider  $n$  robots positioned on the vertices of a regular polygon with side length 1. Now take one robot and mirror its position along the line segment connecting its two neighbors (cf. Figure 3). Next, we assume that all robots would move a constant distance along the angle bisector between their direct neighbors in the given gathering protocol. Other movements would lead to the same effect since the robots are disoriented. In the given configuration,  $n - 1$  robots would



■ **Figure 2** Ideally, every robot that is close to the boundary of the global convex hull would move a constant distance inwards.



■ **Figure 3** Visualization of the example to emphasize that continuous protocols cannot be directly translated to the LCM case.

move a constant distance inside the global convex hull while one robot even leaves the global convex hull. Not only that the global convex hull does not decrease as desired, but also the connectivity of  $UBG(t)$  is not maintained as the robot moving outside loses connectivity to its direct neighbors. Consequently, discrete gathering protocols have to move the robots more carefully to maintain the connectivity of  $UBG(t)$ .

### 3.1 $\lambda$ -contracting Protocols

Initially, we emphasize two core features of the protocols. A discrete protocol is *connectivity preserving* if it always maintains the connectivity of  $UBG(t)$ . Due to the limited visibility and disorientation, every protocol to solve GATHERING and NEAR-GATHERING must be connectivity preserving since it is deterministically impossible to reconnect lost robots to the remaining swarm. Moreover, we study protocols that are *invariant*, i.e., the movement of a robot does not change no matter how its local coordinate system is oriented<sup>3</sup>. This is a natural assumption since the robots have variable disorientation and thus cannot rely on their local coordinate system to synchronize their movement with nearby robots. Moreover, many known protocols under the given robot capabilities are invariant, e.g., [3, 13, 39, 40].

► **Definition 1.** Let  $Q$  be a convex polytope with diameter  $diam$  and  $0 < \lambda \leq 1$  a constant. A point  $p \in Q$  is called to be  **$\lambda$ -centered** if it is the midpoint of a line segment that is completely contained in  $Q$  and has a length of  $\lambda \cdot diam$ .

► **Definition 2.** A connectivity preserving and invariant discrete robot formation protocol  $\mathcal{P}$  is called  **$\lambda$ -contracting** if  $target_i^{\mathcal{P}}(t)$  is a  $\lambda$ -centered point of  $hull_i^t$  for every robot  $r_i$  and every  $t \in \mathbb{N}_0$ .

Two examples of  $\lambda$ -centered points are depicted in Figure 1 (contained in Section 1.2). Observe that Definition 2 does not necessarily enforce a final gathering of the protocols. Consider, for instance, two robots. A protocol that demands the two robots to move halfway towards the midpoint between themselves would be  $1/4$ -contracting, but the robots would only *converge* towards the same position. However, for GATHERING, the robots must compute the same target point eventually. We demand this by requiring that there is a constant  $c < 1$ , such that  $N_i(t) = N_j(t)$  and  $diam_i(t) = diam_j(t) \leq c$  implies that the robots compute the same target point. Protocols that have this property are called *collapsing*. Observe

<sup>3</sup> Note that the protocols do not need to be invariant to ensure GATHERING. Nevertheless, being invariant becomes important when we study the NEAR-GATHERING problem. For ease of description, we consider invariant protocols in general.



that being collapsing is reasonable since  $\lambda$ -contracting demands that robots compute target points inside their local convex hulls and hence, the robots' local diameters are monotonically decreasing in case no further robot enters their neighborhood. Hence, demanding a threshold to enforce moving to the same point is necessary to ensure a final gathering. For ease of description, we fix  $c = 1/2$  in this work. However,  $c$  could be chosen as an arbitrary constant by scaling the obtained runtime bounds with a factor of  $1/c$ .

► **Definition 3.** *A discrete robot formation protocol  $\mathcal{P}$  is a  $\lambda$ -contracting gathering protocol if  $\mathcal{P}$  is  $\lambda$ -contracting and collapsing.*

### 3.2 Analysis of $\lambda$ -contracting Gathering Protocols

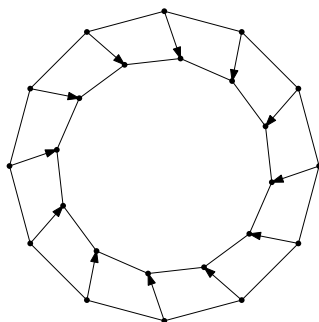
In the following, we state upper and lower bounds about  $\lambda$ -contracting gathering protocols. When considering  $d = 1$ ,  $\lambda$ -contracting gathering protocols are optimal.

► **Theorem 4.** *Consider a swarm of robots in  $\mathbb{R}$ . Every  $\lambda$ -contracting gathering protocol gathers all robots in  $\Theta(\Delta)$  rounds.*

The proof of Theorem 4 can be found in the full version of this paper [14]. For larger dimensions, we start with a lower bound that holds for a larger class of protocols but is especially valid for  $\lambda$ -contracting gathering protocols. The lower bound holds for all discrete gathering protocols that compute robot target points always inside local convex hulls. The proof of the lower bound is in most parts identical to the lower bound of the GTC protocol [24]. Essentially, we prove that, in the configuration where all robots are located on the vertices of a regular polygon with side length 1, GTC is the best possible of all protocols that compute target points inside of local convex hulls.

► **Theorem 5.** *For a swarm of  $n$  robots in  $\mathbb{R}^d$  with  $d \geq 2$  and diameter  $\Delta$  there exists an initial configuration such that every discrete gathering protocol  $\mathcal{P}$  that ensures  $\text{target}_i^{\mathcal{P}}(t) \in \text{hull}_i^t$  for all robots  $r_i$  and all rounds  $t \in \mathbb{N}_0$ , requires  $\Omega(\Delta^2)$  rounds to gather all robots.*

**Proof.** In the following, we assume  $n \geq 5$ . Consider  $n$  robots that are located on the vertices of a regular polygon with side length 1. Observe first that due to the disorientation and because the protocols are deterministic, the local coordinate systems of the robots could be chosen such that the configuration remains a regular polygon forever (see Figure 4 for an example).



■ **Figure 4** Initially, the robots are located on the surrounding regular polygon. The local coordinate systems of the robots can be chosen such that all robots execute the same movement in a rotated fashion such that the configuration remains a regular polygon (depicted by the inner regular polygon).

## 15:10 Efficient (Near)-Gathering of Disoriented Robots with Limited Visibility

Henceforth, we assume in the following that the robots remain on the vertices of a regular polygon. Let  $C$  be the surrounding circle and  $r_C$  its radius. For large  $n$ , the circumference  $p_C$  of  $C$  is  $\approx n$  and  $r_C \approx \frac{n}{2\pi}$ . Hence,  $\Delta \approx \frac{n}{\pi}$ . We show that any  $\lambda$ -contracting protocol (not only gathering protocols) requires  $\Omega(\Delta^2)$  rounds until  $p_C \leq \frac{2}{3}n$ . As long as  $p_C \geq \frac{2}{3}n$ , each robot can observe exactly two neighbors at distance  $\frac{2}{3} \leq s \leq 1$ .

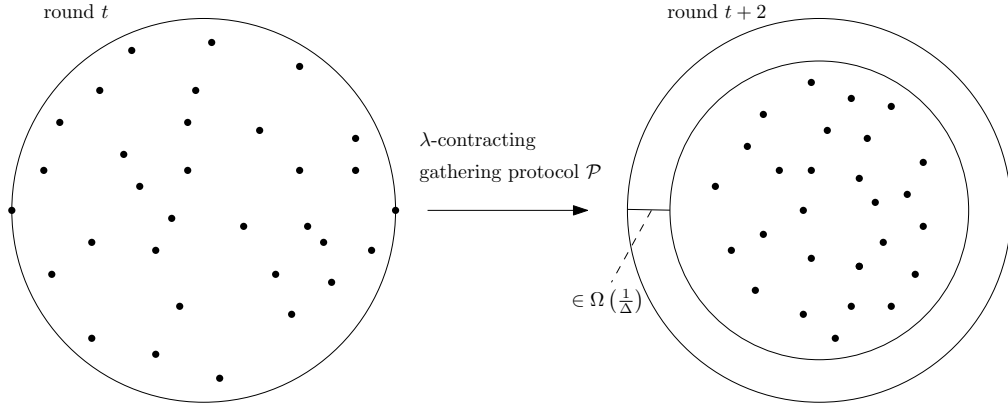
The internal angles of a regular polygon have a size of  $\gamma = \frac{(n-2) \cdot \pi}{n}$ . Fix any robot  $r_i$  and assume that  $p_i(t) = (0, 0)$  and the two neighbors are at  $p_{i-1}(t) = (-s \cdot \sin(\frac{\gamma}{2}), s \cdot \cos(\frac{\gamma}{2}))$  and  $p_{i+1}(t) = p_{i-1}(t) = (s \cdot \sin(\frac{\gamma}{2}), s \cdot \cos(\frac{\gamma}{2}))$ . Now, consider the target point  $\text{target}_i^{\mathcal{P}}(t) = (x_{\text{target}_i^{\mathcal{P}}(t)}, y_{\text{target}_i^{\mathcal{P}}(t)})$ . Observe that the radius  $r_C$  decreases by exactly  $y_{\text{target}_i^{\mathcal{P}}(t)}$ . Next, we derive an upper bound on  $y_{\text{target}_i^{\mathcal{P}}(t)}$ :  $y_{\text{target}_i^{\mathcal{P}}(t)} = s \cdot \cos(\frac{\gamma}{2}) \leq \cos(\frac{\gamma}{2}) = \cos(\frac{(n-2) \cdot \pi}{2n})$ .

Now, we use  $\cos(x) \leq -x + \frac{\pi}{2}$  for  $0 \leq x \leq \frac{\pi}{2}$ . Hence, we obtain  $\cos(\frac{(n-2) \cdot \pi}{2n}) \leq -\frac{(n-2) \cdot \pi}{2n} + \frac{\pi}{2} = -\frac{\pi}{2} + \frac{\pi}{n} + \frac{\pi}{2} = \frac{\pi}{n}$ . Therefore, it takes at least  $\frac{n^2}{3}$  rounds until  $r_C$  has decreased by at least  $\frac{n}{3}$ . The same holds for the perimeter. All in all, it takes at least  $\frac{n^2}{3} \in \Omega(\Delta^2)$  rounds until the  $r_C$  decreases by at least  $\frac{n}{3}$ . ◀

We state a matching upper bound for  $\lambda$ -contracting protocols in two dimensions (later for any  $d \geq 2$ ). We first focus on robots in the Euclidean plane to make the core ideas visualizable.

► **Theorem 6.** *Consider a swarm of robots in  $\mathbb{R}^2$  with diameter  $\Delta$ . Every  $\lambda$ -contracting gathering protocol gathers all robots in  $\frac{171 \cdot \pi \cdot \Delta^2}{\lambda^3} + 1 \in \mathcal{O}(\Delta^2)$  rounds.*

**High-Level Description.** The proof is inspired by the proof of the GTC protocol [24]. The proof aims to show that the radius of the global smallest enclosing circle (SEC), i.e., the SEC that encloses all robots' positions in a global coordinate system, decreases by  $\Omega(1/\Delta)$  every two rounds. Since the initial radius is upper bounded by  $\Delta$ , the runtime of  $\mathcal{O}(\Delta^2)$  follows. See Figure 5 for a visualization.



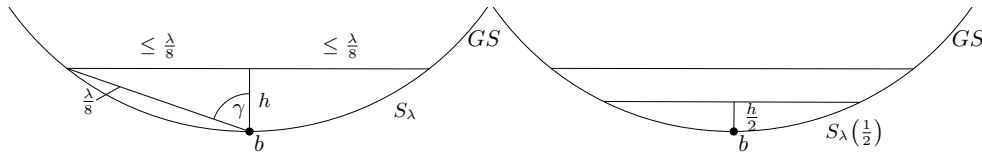
■ **Figure 5** We show that the radius of the global SEC decreases by  $\Omega(1/\Delta)$  every two rounds.

We consider the fixed circular segment  $S_\lambda$  of the global SEC and analyze how the inside robots behave. A circular segment is a region of a circle “cut off” by a chord. The circular segment  $S_\lambda$  has a chord length of at most  $\lambda/4$  (for a formal definition, see below) and we can prove a height  $h$  of  $S_\lambda$  in the order of  $\Omega(1/\Delta)$  (Lemma 8). Observe that in any circular segment, the chord’s endpoints are the points that have a maximum distance within the circular segment, and hence, the maximum distance between any pair of points in  $S_\lambda$  is

at most  $\lambda/4$ . Now, we split the robots inside of  $S_\lambda$  into two classes: the robots  $r_i$  with  $\text{diam}_i(t) > 1/4$  and the others with  $\text{diam}_i(t) \leq 1/4$ . Recall that every robot  $r_i$  moves to the  $\lambda$ -centered point  $\text{target}_i^{\mathcal{P}}(t)$ . Moreover,  $\text{target}_i^{\mathcal{P}}(t)$  is the midpoint of a line segment  $\ell$  of length  $\lambda \cdot \text{diam}_i(t)$  that is completely contained in the local convex hull of  $r_i$ . For robots with  $\text{diam}_i(t) > 1/4$  we have that  $\ell$  is larger than  $\lambda/4$  and thus,  $\ell$  cannot be completely contained in  $S_\lambda$ . Hence,  $\ell$  either connects two points outside of  $S_\lambda$  or one point inside and another outside. In the former case,  $\text{target}_i^{\mathcal{P}}(t)$  is outside of  $S_\lambda$ , and in the latter case,  $\text{target}_i^{\mathcal{P}}(t)$  is outside of a circular segment with half the height  $h$  of  $S_\lambda$ . See Lemma 9 for a formal statement of the first case.

It remains to argue about robots with  $\text{diam}_i(t) < 1/4$ . Here, we consider a circular segment with an even smaller height, namely  $h \cdot \lambda/4$ . We will see that all robots which compute a target point inside this circular segment (which can only be robots with  $\text{diam}_i(t) < \lambda/4$ ) will move exactly to the same position. Hence, in round  $t + 1$  there is only one position in the circular segment with height  $h \cdot \lambda/4$  occupied by robots. All other robots are located outside of the circular segment with height  $h/2$ . As a consequence, for all robots  $r_i$  in the circular segment with height  $h \cdot \lambda/4$ , it must hold  $\text{target}_i^{\mathcal{P}}(t)$  is outside of the circular segment with height  $h \cdot \lambda/4$ . See Lemma 10 for a formal statement. Finally, Lemma 11 combines the previous statements and gives a lower bound on how much the radius of the global SEC decreases.

**Detailed Analysis.** First, we introduce some definitions. Let  $GS := GS(t)$  be the (global) smallest enclosing circle of all robots in round  $t$  and  $R := R(t)$  its radius. Now, fix any point  $b$  on the boundary of  $GS$ . The two points in distance  $\lambda/8$  of  $b$  on the boundary of  $GS$  determine the circular segment  $S_\lambda$  with height  $h$ . In the following, we determine by  $S_\lambda(c)$  for  $0 < c \leq 1$  the circular segment with height  $c \cdot h$  that is contained in  $S_\lambda$ . See Figure 6 for a depiction of the circular segments  $S_\lambda$  and  $S_\lambda(1/2)$  (that is used in the proofs). In the following, all lemmata consider robots that move according to a  $\lambda$ -contracting gathering protocol  $\mathcal{P}$ .



■ **Figure 6** The circular segments  $S_\lambda$  (to the left) and  $S_\lambda(1/2)$  of the global SEC  $GS$  are depicted.

In the following, we prove that all robots leave the circular segment  $S_\lambda(\lambda/4)$  every two rounds. As a consequence, the radius of  $GS$  decreases by at least  $\lambda/4 \cdot h$ . Initially, we give a bound on  $h$ . We use Jung’s Theorem (Theorem 7) to obtain a bound on  $R$  and also on  $h$ .

► **Theorem 7** (Jung’s Theorem [34, 35]). *The smallest enclosing hypersphere of a point set  $K \subset \mathbb{R}^d$  with diameter  $\text{diam}$  has a radius of at most  $\text{diam} \cdot \sqrt{\frac{d}{2 \cdot (d+1)}}$ .*

► **Lemma 8.**  $h \geq \frac{\sqrt{3} \cdot \lambda^2}{64\pi\Delta}$ .

**Proof.** Initially, we give an upper bound on the angle  $\gamma$ , see Figure 6 for its definition. The circumference of  $GS$  is  $2\pi R$ . We can position at most  $\frac{16}{\lambda}\pi R$  points on the boundary of  $GS$  that are at distance  $\frac{\lambda}{8}$  from the points closest to them and form a regular convex polygon. The internal angle of this regular polygon is  $2\gamma$ . Hence, the sum of all internal angles is

## 15:12 Efficient (Near)-Gathering of Disoriented Robots with Limited Visibility

$(\frac{16}{\lambda}\pi R - 2) \cdot \pi$ . Thus, each individual angle has a size of at most  $\frac{(\frac{16}{\lambda}\pi R - 2) \cdot \pi}{\frac{16}{\lambda}\pi R} = \pi - \frac{2\pi}{\frac{16}{\lambda}\pi R} = \pi - \frac{\lambda}{8R}$ . Hence,  $\gamma \leq \frac{\pi}{2} - \frac{\lambda}{16R}$ . Now, we are able to bound  $h$ . First of all, we derive a relation between  $h$  and  $\gamma$ :  $\cos(\gamma) = \frac{h}{\frac{\lambda}{8}} = \frac{8h}{\lambda} \iff h = \frac{\lambda \cdot \cos(\gamma)}{8}$ . In the following upper bound, we make use of the fact that  $\cos(x) \geq -\frac{2}{\pi}x + 1$  for  $x \in [0, \frac{\pi}{2}]$ .

$$h = \frac{\lambda \cdot \cos(\gamma)}{8} \geq \frac{\lambda \cdot \cos\left(\frac{\pi}{2} - \frac{\lambda}{16R}\right)}{8} \geq \frac{\lambda \cdot \left(-\frac{2}{\pi} \cdot \left(\frac{\pi}{2} - \frac{\lambda}{16R}\right) + 1\right)}{8} = \frac{\lambda \cdot \frac{\lambda}{8\pi R}}{8} = \frac{\lambda^2}{64\pi R}$$

Applying Theorem 7 with  $d = 2$  yields  $h \geq \frac{\sqrt{3} \cdot \lambda^2}{64\pi\Delta}$ .  $\blacktriangleleft$

We continue to prove that all robots leave  $S_\lambda(\lambda/4)$  every two rounds. First of all, we analyze robots for which  $\text{diam}_i(t) > 1/4$ . These robots even leave the larger circular segment  $S_\lambda(1/2)$ .

► **Lemma 9.** *For any robot  $r_i$  with  $\text{diam}_i(t) > 1/4$ :  $\text{target}_i^{\mathcal{P}}(t) \in GS \setminus S_\lambda(1/2)$ .*

**Proof.** Since  $\text{diam}_i(t) > 1/4$  and  $\mathcal{P}$  is  $\lambda$ -contracting,  $\text{target}_i^{\mathcal{P}}(t)$  is the midpoint of a line segment  $\ell_i^{\mathcal{P}}(t)$  of length at least  $\lambda \cdot \text{diam}_i(t) > \lambda/4$ . As the maximum distance between any pair of points inside of  $S_\lambda$  is  $\frac{\lambda}{4}$ , it follows that  $\ell_i^{\mathcal{P}}(t)$  either connects two points outside of  $S_\lambda$  or one point inside and another point outside. In the first case,  $\text{target}_i^{\mathcal{P}}(t)$  lies outside of  $S_\lambda$  (since the maximum distance between any pair of points inside of  $S_\lambda$  is  $\frac{\lambda}{4} \leq 1/4 < \text{diam}_i(t)$ ). In the second case,  $\text{target}_i^{\mathcal{P}}(t)$  lies outside of  $S_\lambda(1/2)$  since, in the worst case, one endpoint of  $\ell_i^{\mathcal{P}}(t)$  is the point  $b$  used in the definition of  $GS$  (see the beginning of Section 3.2) and the second point lies very close above of  $S_\lambda(1/2)$ . Since  $\text{target}_i^{\mathcal{P}}(t)$  is the midpoint of  $\ell_i^{\mathcal{P}}(t)$ , it lies closely above of  $S_\lambda(1/2)$ . Every other position of the two endpoints of  $\ell_i^{\mathcal{P}}(t)$  would result in a point  $\text{target}_i^{\mathcal{P}}(t)$  that lies even farther above of  $S_\lambda(1/2)$ .  $\blacktriangleleft$

Now, we consider the case of a single robot in  $S_\lambda(\lambda/4)$ , and its neighbors are located outside of  $S_\lambda(1/2)$ . We prove that this robot leaves  $S_\lambda(\lambda/4)$ . Additionally, we prove that none of the robots outside of  $S_\lambda(1/2)$  that see the single robot in  $S_\lambda(\lambda/4)$  enters  $S_\lambda(\lambda/4)$ .

► **Lemma 10.** *Consider a robot  $r_i$  located in  $S_\lambda(\lambda/4)$ . If all its neighbors are located outside of  $S_\lambda(1/2)$ ,  $\text{target}_i^{\mathcal{P}}(t) \in GS \setminus S_\lambda(\lambda/4)$ . Similarly, for a robot  $r_i$  that is located outside of  $S_\lambda(1/2)$  and that has only one neighbor located in  $S_\lambda(\lambda/4)$ ,  $\text{target}_i^{\mathcal{P}}(t) \in GS \setminus S_\lambda(\lambda/4)$ .*

**Proof.** First, we consider a robot  $r_i$  that is located in  $S_\lambda(\lambda/4)$  and all its neighbors are above of  $S_\lambda(1/2)$ . Let  $p_1$  and  $p_2$  be the two points of  $\text{hull}_i^t$  closest to the intersection points of  $\text{hull}_i^t$  and the boundary of  $S_\lambda(1/2)$  ( $p_1$  and  $p_2$  are infinitesimally above of  $S_\lambda(1/2)$ ). In case  $\text{hull}_i^t$  consists of only two robots, define  $p_1$  to be the intersection point of  $\text{hull}_i^t$  and  $S_\lambda(1/2)$  and  $p_2 = p_i(t)$ . The maximum distance  $d_{\max}$  between any pair of points in  $\text{hull}_i^t \cap S_\lambda(1/2)$  is less than  $\max\{|p_1 - p_2|, |p_1 - p_i(t)|, |p_2 - p_i(t)|\}$ , since  $p_1$  and  $p_2$  are slightly above of  $S_\lambda(1/2)$ . Clearly,  $\text{diam}_i(t) \geq d_{\max}$ . Thus, the maximum distance between any pair of points in  $\text{hull}_i^t \cap S_\lambda(\frac{\lambda}{2})$  is less than  $\lambda \cdot d_{\max}$ . We conclude that  $\text{target}_i^{\mathcal{P}}(t)$  must be located above of  $S_\lambda(\lambda/4)$  since  $\text{target}_i^{\mathcal{P}}(t)$  is the midpoint of a line segment of length  $\lambda \cdot \text{diam}_i(t) \geq \lambda \cdot d_{\max}$  either connecting two robots above of  $S_\lambda(\lambda/4)$  or one robot inside of  $S_\lambda(\lambda/4)$  and one robot outside of  $S_\lambda(\lambda/4)$ . The arguments for the opposite case –  $r_i$  is located in  $S_\lambda(1/2)$ , one neighbor of  $r_i$  is located in  $S_\lambda(\lambda/4)$  and all others are also outside of  $S_\lambda(1/2)$  – are analogous.  $\blacktriangleleft$

Next, we derive with help of Lemmas 9 and 10 that the circular segment  $S_\lambda(\lambda/4)$  is empty after two rounds. Additionally, we analyze how much  $R(t)$  decreases.

► **Lemma 11.** *For any round  $t$  with  $\text{diam}(t) \geq 1/2$ ,  $R(t+2) \leq R(t) - \frac{\lambda^3 \cdot \sqrt{3}}{256 \cdot \pi \cdot \Delta}$ .*

**Proof.** Fix any circular  $S_\lambda$  and consider the set of robots  $R_S$  that are located in  $S_\lambda(\lambda/4)$  or compute a target point in  $S_\lambda(\lambda/4)$ . Initially, we argue that all robots in  $R_S$  can see each other. Via Lemma 9, we obtain that for every robot  $r_i \in R_S$  that computes a target point in  $S_\lambda(\lambda/4)$ ,  $\text{diam}_i(t) \leq 1/4$ . Since the maximum distance between any pair of points in  $S_\lambda(\lambda/4)$  is less than  $1/4$  (as the maximum distance of any points in the larger circular segment  $S_\lambda$  is  $\lambda/4$ ), we conclude that, a robot which is not located in  $S_\lambda(\lambda/4)$  but computes its target point inside, is at distance at most  $1/4$  from  $S_\lambda(\lambda/4)$ . Hence, via the triangle inequality, it is located at distance at most  $1/2$  from any other robot in  $R_S$ . Thus, all robots in  $R_S$  can see each other. Now consider the robot  $r_{\min} \in R_S$  which is one of the robots of  $R_S$  with the minimal number of visible neighbors. Furthermore,  $A_{\min}$  is the set of robots that have exactly the same neighborhood as  $r_{\min}$ . For all robots  $r_j \in R_S \setminus A_{\min}$ , we have that  $r_j$  can see  $r_{\min}$  and at least one robot that  $r_{\min}$  cannot see. Thus,  $\text{diam}_j(t) > 1$ . We can conclude with help of Lemma 9 that all robots in  $R_S \setminus A_{\min}$  compute a target point outside of  $S_\lambda(1/2)$ . Since all robots  $r_i \in A_{\min}$  have the same neighborhood and  $\text{diam}_i(t) < 1/4$ , they also compute the same target point ( $\lambda$ -contracting gathering protocols are collapsing). Thus, at the beginning of round  $t+1$ , at most one position in  $S_\lambda(\lambda/4)$  is occupied. In round  $t+1$  we have the picture that one position in  $S_\lambda(\lambda/4)$  is occupied and all neighbors are located above of  $S_\lambda(1/2)$ . Lemma 10 yields that the robots in  $S_\lambda(\lambda/4)$  compute a target point outside. Moreover, Lemma 10 yields as well that no robot outside of  $S_\lambda(\lambda/4)$  computes a target point inside and thus,  $S_\lambda(\lambda/4)$  is empty in round  $t+2$ . Since the circular segment  $S_\lambda$  has been chosen arbitrarily, the arguments hold for the entire circle  $GS$  and thus,  $R(t+2) \leq R(t) - \lambda/4 \cdot h \leq R(t) - \frac{\lambda^3 \cdot \sqrt{3}}{256 \cdot \pi \cdot \Delta}$ . ◀

Finally, we can conclude with help of Lemma 11 the main Theorem 6.

**Proof of Theorem 6.** First, we bound the initial radius of  $GS$ :  $R(0) \leq \Delta/\sqrt{3}$  (Theorem 7). Lemma 11 yields that  $R(t)$  decreases every two rounds by at least  $\frac{\lambda^3 \cdot \sqrt{3}}{256 \cdot \pi \cdot \Delta}$ . Thus, it requires  $2 \cdot \frac{256 \cdot \pi \cdot \Delta}{\lambda^3}$  rounds until  $R(t)$  decreases by at least  $\sqrt{3}$ . Next, we bound how often this can happen until  $R(t) \leq \frac{1}{4}$  and thus  $\text{diam}(t) \leq \frac{1}{2}$ :  $\frac{\Delta}{\sqrt{3}} - x \cdot \sqrt{3} \leq \frac{1}{4} \iff \frac{\Delta}{3} - \frac{1}{4 \cdot \sqrt{3}} \leq x$ .

All in all, it requires  $x \cdot \frac{512 \cdot \pi \cdot \Delta}{\lambda^3} = \left(\frac{\Delta}{3} - \frac{1}{4 \cdot \sqrt{3}}\right) \cdot \frac{512 \cdot \pi \cdot \Delta}{\lambda^3} \leq \frac{171 \cdot \pi \cdot \Delta^2}{\lambda^3}$  rounds until  $\text{diam}(t) \leq \frac{1}{2}$ . As soon as  $\text{diam}(t) \leq \frac{1}{2}$ , all robots can see each other, compute the same target point and will reach it in the next round. ◀

**Upper Bound in  $d$ -dimensions.** The upper bound we derived for two dimensions can also be generalized to every dimension  $d$ . Only the constants in the runtime increase slightly.

► **Theorem 12.** *Consider a team of  $n$  robots located in  $\mathbb{R}^d$ . Every  $\lambda$ -contracting gathering protocol gathers all robots in  $\frac{256 \cdot \pi \cdot \Delta^2}{\lambda^3} + 1 \in \mathcal{O}(\Delta^2)$  rounds.*

### 3.3 Examples of $\lambda$ -contracting Gathering Protocols

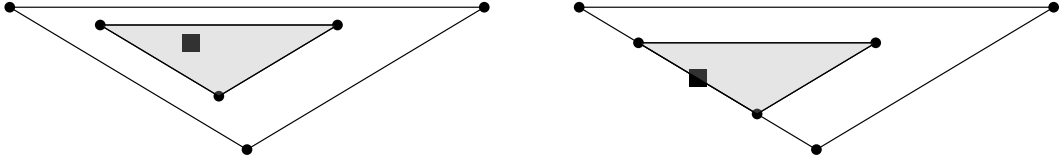
Next, we present examples of  $\lambda$ -contracting gathering protocols. Before introducing the concrete protocols, we describe an important subclass of  $\lambda$ -contracting protocols, denoted as  $(\alpha, \beta)$ -contracting protocols, a powerful tool to decide whether a given protocol is  $\lambda$ -contracting. Afterward, we introduce the known protocol GTC [3] and prove it to be  $\lambda$ -contracting. Additionally, we introduce two further two-dimensional  $\lambda$ -contracting gathering protocols: GTMD and GTCDMB.

**$(\alpha, \beta)$ -contracting Protocols.** While the definition of  $\lambda$ -contracting gathering protocols describes the core properties of efficient protocols to solve GATHERING, it might be practically challenging to determine whether a given protocol is  $\lambda$ -contracting. Concrete protocols often are designed as follows: robots compute a *desired* target point and move as close as possible towards it without losing connectivity [3, 13, 39]. The GTC protocol, for instance, uses this rule. Since the robots do not necessarily reach the desired target point, it is hard to determine whether the resulting point is  $\lambda$ -centered. Therefore, we introduce a two-stage definition:  $(\alpha, \beta)$ -contracting protocols. The parameter  $\alpha$  represents an  $\alpha$ -centered point (Definition 1) and  $\beta$  describes how close the robots move towards the point.

► **Definition 13.** Let  $c_1, \dots, c_k$  with  $c_i \in \mathbb{R}^d$  be the vertices of a convex polytope  $Q$ ,  $p \in Q$  and  $0 < \beta \leq 1$  a constant.  $Q(p, \beta)$  is the convex polytope with vertices  $p + (1 - \beta) \cdot (c_i - p)$ .

Now, we are ready to define the class of  $(\alpha, \beta)$ -contracting protocols. It uses a combination of Definitions 1 and 13: the target points of the robots must be inside of the  $\beta$ -scaled local convex hull around an  $\alpha$ -centered point. See also Figure 7 for a visualization of valid target points in  $(\alpha, \beta)$ -contracting protocols. Recall that  $\text{hull}_i^t$  defines the convex hull of all neighbors of  $r_i$  including  $r_i$  in round  $t$  and  $\text{hull}_i^t(p, \beta)$  is the scaled convex hull around  $p$  (Definition 13).

► **Definition 14.** A connectivity preserving and invariant discrete robot formation protocol  $\mathcal{P}$  is called to be  **$(\alpha, \beta)$ -contracting**, if there exists an  $\alpha$ -centered point  $\alpha\text{-center}_i^{\mathcal{P}}(t)$  s.t.  $\text{target}_i^{\mathcal{P}}(t) \in \text{hull}_i^t(\alpha\text{-center}_i^{\mathcal{P}}(t), \beta)$  for every robot  $r_i$  and every  $t \in \mathbb{N}_0$ . Moreover,  $\mathcal{P}$  is called an  **$(\alpha, \beta)$ -contracting gathering protocol** if  $\mathcal{P}$  is  $(\alpha, \beta)$ -contracting and collapsing.



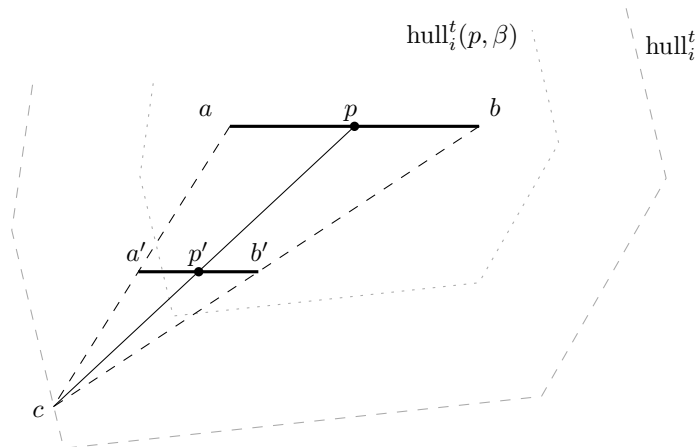
■ **Figure 7** Two examples of valid target points of  $(\alpha, \beta)$ -contracting protocols. The small gray triangle represents the  $\frac{1}{2}$ -scaled convex hull around an  $\frac{1}{4}$ -centered point marked with a square.

Next, we state the relation between  $(\alpha, \beta)$ -contracting and  $\lambda$ -contracting protocols.

► **Theorem 15.** Every  $(\alpha, \beta)$ -contracting protocol  $\mathcal{P}$  is  $\lambda$ -contracting with  $\lambda = \alpha \cdot \beta$ .

**Proof.** From the definition of  $(\alpha, \beta)$ -contracting protocols, we know that for a target point  $\text{target}_i^{\mathcal{P}}(t)$ , there exists a point  $\alpha\text{-center}_i^{\mathcal{P}}(t)$  such that  $\text{target}_i^{\mathcal{P}}(t) \in \text{hull}_i^t(\alpha\text{-center}_i^{\mathcal{P}}(t), \beta)$ . We do the following geometric construction in Figure 8. Let  $p = \alpha\text{-center}_i^{\mathcal{P}}(t)$  and  $p' = \text{target}_i^{\mathcal{P}}(t)$ . We draw a line segment from  $\alpha\text{-center}_i^{\mathcal{P}}(t)$  through  $\text{target}_i^{\mathcal{P}}(t)$  to the boundary of  $\text{hull}_i^t$ . Let  $c$  be the endpoint of this line segment. Because  $p$  is  $\alpha$ -centered, there exists a line segment with length  $\text{diam}_i(t) \cdot \alpha$  through  $p$ , let this be the line segment  $\overline{ab}$ . The line segment  $\overline{a'b'}$  is a parallel to  $\overline{ab}$  inside the triangle  $\Delta_{abc}$ . We know that  $p' \in \text{hull}_i^t(p, \beta)$ , therefore  $|\overline{cp'}| \geq \beta|\overline{cp}|$ . By the intercept theorem, it follows that  $|\overline{a'b'}| \geq \beta|\overline{ab}| = \beta \cdot \alpha \cdot \text{diam}_i(t)$ . Because the points  $a, b$  and  $c$  are all inside  $\text{hull}_i^t$ , the entire triangle  $\Delta_{abc}$  and  $\overline{a'b'}$  are inside  $\text{hull}_i^t$  as well. Therefore,  $\text{target}_i^{\mathcal{P}}(t)$  is a  $\lambda$ -centered point with  $\lambda = \alpha \cdot \beta$ . ◀

**Go-To-The-Center.** As a first example, we study the two-dimensional GTC protocol [3]. It is already known that it gathers all robots in  $\mathcal{O}(n + \Delta^2)$  rounds [24]. We show that GTC is  $(\alpha, \beta)$ -contracting (hence also  $\lambda$ -contracting) and thus, obtain an improved upper runtime



■ **Figure 8** The construction used in the proof of Theorem 15.

bound of  $\mathcal{O}(\Delta^2)$ . The formal description of the GTC protocol can be found in [14]. Robots always move towards the center of the smallest enclosing circle of their neighborhood. To maintain connectivity, *limit circles* are used. Each robot  $r_i$  always stays within the circle of radius  $1/2$  centered in the midpoint  $m_j$  of every visible robot  $r_j$ . Since each robot  $r_j$  does the same, it is ensured that two visible robots always stay within a circle of radius  $1/2$  and thus, they remain connected. Consequently, robots move only that far towards the center of the smallest enclosing circle such that no limit circle is left.

► **Theorem 16.** *GTC is  $(\sqrt{3}/8, 1/2)$ -contracting.*

GTC can be generalized to  $d$ -dimensions by moving robots toward the center of the smallest enclosing hypersphere of their neighborhood. We denote the resulting protocol by  $d$ -GTC, a complete description is deferred to [14].

► **Theorem 17.**  *$d$ -GTC is  $(\sqrt{2}/8, 1/2)$ -contracting.*

**Go-To-The-Middle-Of-The-Diameter (GtMD).** Next, we describe a second two-dimensional protocol that is also  $(\alpha, \beta)$ -contracting. The intuition is quite simple: a robot  $r_i$  moves towards the midpoint of the two robots defining  $\text{diam}_i(t)$ . Similar to the GTC protocol, connectivity is maintained with the help of limit circles. A robot only moves that far towards the midpoint of the diameter such that no limit circle (a circle with radius  $1/2$  around the midpoint of  $r_i$  and each visible robot  $r_j$ ) is left. Observe further that the midpoint of the diameter is not necessarily unique. To make GtMD in cases where the midpoint of the diameter is not unique deterministic, robots move according to GTC. The formal description can be found in [14]. We prove the following property about GtMD.

► **Theorem 18.** *In rounds, where the local diameter of all robots is unique, GtMD is  $(1, 1/10)$ -contracting  $((\sqrt{3}/8, 1/2)$ -contracting otherwise).*

**Go-To-The-Center-Of-The-Diameter-MinBox (GtCDB).** Lastly, we derive a third protocol for robots in  $\mathbb{R}^2$  that is also  $(\alpha, \beta)$ -contracting. It is based on the local *diameter minbox* defined as follows. The local coordinate system is adjusted such that the two robots that define the diameter are located on the  $y$ -axis, and the midpoint of the diameter coincides with the origin. Afterwards, the maximal and minimal  $x$ -coordinates  $x_{\max}$  and  $x_{\min}$  of other

visible robots are determined. Finally, the robot moves towards  $(\frac{1}{2} \cdot (x_{\min} + x_{\max}), 0)$ . The box boundaries with  $x$ -coordinates  $x_{\min}, x_{\max}$  and  $y$ -coordinates  $-\text{diam}_i(t)/2$  and  $\text{diam}_i(t)/2$  is called the *diameter minbox* of  $r_i$ . Note that, similar to GTMD, the diameter minbox of  $r_i$  might not be unique. In this case, a fallback to GTC is used. The complete description of GTCDMB is contained in [14]. Also, GTCDMB is  $(\alpha, \beta)$ -contracting.

► **Theorem 19.** *In rounds, where the local diameter of all robots is unique, GTCDMB is  $(\sqrt{3}/8, 1/10)$ -contracting  $((\sqrt{3}/8, 1/2)$ -contracting otherwise).*

## 4 Collision-free Near-Gathering Protocols

In this section, we study the NEAR-GATHERING problem for robots located in  $\mathbb{R}^d$  under the  $\mathcal{S}$ SYNC scheduler. The main difference to GATHERING is that robots may never collide (move to the same position). We introduce a very general approach to NEAR-GATHERING that builds upon  $\lambda$ -contracting gathering protocols (Section 3). We show how to transform *any*  $\lambda$ -contracting gathering protocol into a *collision-free*  $\lambda$ -contracting protocol that solves NEAR-GATHERING in  $\mathcal{O}(\Delta^2)$  epochs under the  $\mathcal{S}$ SYNC scheduler. The only difference in the robot model (compared to GATHERING in Section 3) is that we need a slightly stronger assumption on the connectivity: the connectivity range must be by an additive constant smaller than the viewing range. More formally, the connectivity range is 1 while robots have a viewing range of  $1 + \tau$  for a constant  $0 < \tau \leq 2/3$ . Note that the upper bound on  $\tau$  is only required because  $\tau/2$  also represents the maximum movement distance of a robot (see below). In general, the viewing range could also be chosen larger than  $1 + \tau$  without any drawbacks while keeping the maximum movement distance at  $\tau/2$ .

The main idea of our approach can be summarized as follows: first, robots compute a *potential* target point based on a  $\lambda$ -contracting gathering protocol  $\mathcal{P}$  that considers only robots at a distance at most 1. Afterward, a robot  $r_i$  uses the viewing range of  $1 + \tau$  to determine whether its potential target point collides with any potential target point of a nearby neighbor. If there might be a collision,  $r_i$  does not move to its potential target point. Instead, it only moves to a point between itself and the potential target point where no other robot moves to. At the same time, it is also ensured that  $r_i$  moves sufficiently far towards the potential target point to maintain the time bound of  $\mathcal{O}(\Delta^2)$  epochs. To realize the ideas with a viewing range of  $1 + \tau$ , we restrict the maximum movement distance of any robot to  $\tau/2$ . More precisely, if the potential target point of any robot given by  $\mathcal{P}$  is at a distance of more than  $\frac{\tau}{2}$ , the robot moves at most  $\frac{\tau}{2}$  towards it. With this restriction, each robot could only collide with other robots at a distance of at most  $\tau$ . The viewing range of  $1 + \tau$  allows computing the potential target point based on  $\mathcal{P}$  of all neighbors at a distance at most  $\tau$ . By knowing all these potential target points, the own target point of the collision-free protocol can be chosen. While this only summarizes the key ideas, we give a more technical intuition and a summary of the proof in Section 4.2.

► **Theorem 20.** *For every  $\lambda$ -contracting gathering protocol  $\mathcal{P}$ , there exists a collision-free  $\lambda$ -contracting protocol  $\mathcal{P}^{cl}$  which solves NEAR-GATHERING in  $\mathcal{O}(\Delta^2)$  epochs under the  $\mathcal{S}$ SYNC scheduler. Let 1 be the viewing and connectivity range of  $\mathcal{P}$ .  $\mathcal{P}^{cl}$  has a connectivity range of 1 and viewing range of  $1 + \tau$  for a constant  $0 < \tau \leq 2/3$ .*

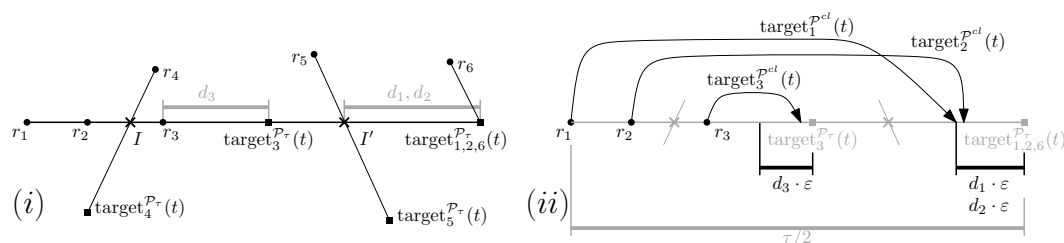
### 4.1 Collision-free Protocol

The construction of the collision-free protocol  $\mathcal{P}^{cl}(\mathcal{P}, \tau, \varepsilon)$  depends on several parameters that we briefly define.  $\mathcal{P}$  is a  $\lambda$ -contracting gathering protocol (designed for robots with a viewing range of 1). The constant  $\tau$  has two purposes. The robots have a viewing range



of  $1 + \tau$  and  $\tau/2$  is the maximum movement distance of any robot,  $0 < \tau \leq 2/3$ . Lastly, the constant  $\varepsilon \in (0, 1/2)$  determines how close each robot moves towards its target point based on  $\mathcal{P}$ . To simplify the notation, we usually write  $\mathcal{P}^{cl}$  instead of  $\mathcal{P}^{cl}(\mathcal{P}, \tau, \varepsilon)$ . Subsequently, we formally define  $\mathcal{P}^{cl}(\mathcal{P}, \tau, \varepsilon)$ . The description is split into three parts that can be found in Algorithms 1–3. The main routine is contained in Algorithm 1. The other two Algorithms 2 and 3 are used as subroutines.

The computation of  $\text{target}_i^{\mathcal{P}^{cl}}(t)$  is based on the movement  $r_i$  would do in a slightly modified version of  $\mathcal{P}$ , denoted as  $\mathcal{P}_\tau$ . The protocol  $\mathcal{P}_\tau$  is defined in Algorithm 3 and a detailed intuition of why it is needed can be found in Section 4.2. The position of  $\text{target}_i^{\mathcal{P}^{cl}}(t)$  lies on the *collision vector*  $\text{colvec}_i^{\mathcal{P}_\tau}(t)$ , the vector from  $p_i(t)$  to  $\text{target}_i^{\mathcal{P}_\tau}(t)$ . On  $\text{colvec}_i^{\mathcal{P}_\tau}(t)$ , there may be several *collision points*. These are either current positions, potential target points ( $\text{target}_k^{\mathcal{P}_\tau}(t)$ ) of other robots  $r_k$  or single intersection points between  $\text{colvec}_i^{\mathcal{P}_\tau}(t)$  and another collision vector  $\text{colvec}_k^{\mathcal{P}_\tau}(t)$ . The computation of collision points is defined in Algorithm 2. Moreover,  $d_i > 0$  is the minimal distance between a collision point and  $\text{target}_i^{\mathcal{P}_\tau}(t)$ . The final target point  $\text{target}_i^{\mathcal{P}^{cl}}(t)$  is exactly at distance  $d_i \cdot \varepsilon \cdot 2/\tau \cdot |\text{colvec}_i^{\mathcal{P}_\tau}(t)|$  from  $\text{target}_i^{\mathcal{P}_\tau}(t)$ . Figure 9 gives an example of collision points and target points of  $\mathcal{P}^{cl}$ .



■ **Figure 9** Example of  $\text{target}_i^{\mathcal{P}^{cl}}(t)$  with  $\tau = 2/3$  and  $\varepsilon = 0.49$ . (i) shows the collision points and computation of  $d_1, d_2$  and  $d_3$  (line 3 in Algorithm 1). (ii) shows the positions where  $r_1, r_2$  and  $r_3$  will move to in protocol  $\mathcal{P}^{cl}$  as returned by Algorithm 1.

■ **Algorithm 1**  $\text{target}_i^{\mathcal{P}^{cl}(\mathcal{P}, \tau, \varepsilon)}(t)$ .

- 
- 1:  $R_i \leftarrow \{r_k : |p_k(t) - p_i(t)| \leq \tau\}$  ▷ Robots in radius  $\tau$  around  $r_i$  (including  $r_i$ )
  - 2:  $C_i \leftarrow \text{collisionPoints}_i^{\mathcal{P}_\tau}(R_i, t)$  ▷ Collision points on  $\text{colvec}_i^{\mathcal{P}_\tau}(t)$ , see Algorithm 2
  - 3:  $d_i \leftarrow \min \left( \left\{ |c - \text{target}_i^{\mathcal{P}_\tau}(t)| : c \in C_i \setminus \{\text{target}_i^{\mathcal{P}_\tau}(t)\} \right\} \right)$  ▷ min. dist. to collision point
  - 4: **return** point on  $\text{colvec}_i^{\mathcal{P}_\tau}(t)$  with distance  $d_i \cdot \varepsilon \cdot 2/\tau \cdot |\text{colvec}_i^{\mathcal{P}_\tau}(t)|$  to  $\text{target}_i^{\mathcal{P}_\tau}(t)$
- 

## 4.2 Proof Summary and Intuition

In the following, we describe the technical intuitions behind the protocol  $\mathcal{P}^{cl}$ . Since the intuition is closely interconnected with the formal analysis, we also give a proof outline here. The proofs of all stated lemmas and theorems can be found in [14]. The entire protocol  $\mathcal{P}^{cl}$  is described in Section 4.1. The idea for  $\mathcal{P}^{cl}$  is straightforward: robots compute a potential target point based on a  $\lambda$ -contracting gathering protocol  $\mathcal{P}$  (that uses a viewing range of 1), restrict the maximum movement distance to  $\tau/2$  and use the viewing range of  $1 + \tau$  to avoid collisions with robots in the distance at most  $\tau$ . However, there are several technical details we want to emphasize in this section.

---

**Algorithm 2** collisionPoints $_i^{\mathcal{P}}(R_i, t)$ .

---

```

1:  $C_i \leftarrow$  empty set
2: for all  $r_k \in R_i$  do
3:   compute target $_k^{\mathcal{P}}(t)$  and colvec $_k^{\mathcal{P}}(t)$  in local coordinate system of  $r_i$ 
4:   if  $p_k(t) \in \text{colvec}_k^{\mathcal{P}}(t)$  then
5:     add  $p_k(t)$  to  $C_i$   $\triangleright$  position of  $r_k$ 
6:   if target $_k^{\mathcal{P}}(t) \in \text{colvec}_i^{\mathcal{P}}(t)$  then
7:     add target $_k^{\mathcal{P}}(t)$  to  $C_i$ 
8:   if colvec $_k^{\mathcal{P}}(t)$  intersects colvec $_i^{\mathcal{P}}(t)$  and is not collinear to colvec $_i^{\mathcal{P}}(t)$  then
9:     add intersection point between colvec $_k^{\mathcal{P}}(t)$  and colvec $_i^{\mathcal{P}}(t)$  to  $C_i$ 
10: return  $C_i$ 

```

---

**Algorithm 3** target $_i^{\mathcal{P}_\tau}(t)$ .

---

```

1: if robots in range 1 have pairwise distance  $\leq \tau/2$  then
2:    $\mathcal{P}^{1+\tau/2} \leftarrow$  protocol  $\mathcal{P}$  scaled to viewing range  $1 + \tau/2$ 
3:    $P_i \leftarrow$  target $_i^{\mathcal{P}^{1+\tau/2}}(t)$ 
4: else
5:    $P_i \leftarrow$  target $_i^{\mathcal{P}}(t)$ 
6: if distance  $p_i(t)$  to  $P_i > \tau/2$  then
7:   return point with distance  $\tau/2$  to  $p_i(t)$  between  $p_i(t)$  and  $P_i$ 
8: else
9:   return  $P_i$ 

```

---

For the correctness and the runtime analysis of the protocol  $\mathcal{P}^{cl}$ , we would like to use the insights into  $\lambda$ -contracting protocols derived in Section 3. However, since the robots compute their potential target point based on a  $\lambda$ -contracting gathering protocol  $\mathcal{P}$  with viewing range 1, this point must not necessarily be  $\lambda$ -centered concerning the viewing range of  $1 + \tau$ . We discuss this problem in more detail in Section 4.3 and motivate the *intermediate* protocol  $\mathcal{P}_\tau$  that is  $\lambda$ -contracting with respect to the viewing range of  $1 + \tau$ .  $\mathcal{P}_\tau$  is only an intermediate protocol since robots still may collide. Afterward in Section 4.4, we explain how to implement collision avoidance and transform the intermediate protocol  $\mathcal{P}_\tau$  into the collision-free protocol  $\mathcal{P}^{cl}$ . Lastly, in Section 4.5 we argue that our transformed and collision-free protocol is still  $\lambda$ -contracting and derive the running time from this property.

### 4.3 The protocol $\mathcal{P}_\tau$

Recall that the main goal is to compute potential target points based on a  $\lambda$ -contracting gathering protocol  $\mathcal{P}$  with viewing range 1. Unfortunately, a direct translation of the protocol loses the  $\lambda$ -contracting property in general. Consider the following example which is also depicted in Figure 10. Assume there are the robots  $r_1, r_2, r_3$  and  $r_4$  in one line with respective distances of  $1/n, 1 + 1/n$  and  $1 + \tau$  to  $r_1$ . It can easily be seen, that the target point target $_1^{\mathcal{P}}(t)$  (protocol  $\mathcal{P}$  has only a viewing range of 1) is between  $r_1$  and  $r_2$ . Such a target point can never be  $\lambda$ -centered with  $\lambda > 2/n$  for  $\mathcal{P}^{cl}$  (with viewing range  $1 + \tau$ ).

Next, we argue how to transform the protocol  $\mathcal{P}$  with viewing range 1 into a protocol  $\mathcal{P}_\tau$  with viewing range  $1 + \tau$  such that  $\mathcal{P}_\tau$  is  $\lambda$ -contracting gathering protocol. The example above already emphasizes the main problem: robots can have very small local diameters  $\text{diam}_i(t)$ . Instead of moving according to  $\mathcal{P}$ , those robots compute a target point based

on  $\mathcal{P}^{1+\tau/2}$ , which is a  $\lambda$ -contracting gathering protocol concerning the viewing range of  $1 + \tau/2$ . Protocol  $\mathcal{P}^{1+\tau/2}$  is obtained by scaling  $\mathcal{P}$  to the larger viewing range of  $1 + \tau/2$ . More precisely, robots  $r_i$  with  $\text{diam}_i(t) \leq \tau/2$  compute their target points based on  $\mathcal{P}^{1+\tau/2}$  and all others according to  $\mathcal{P}$ . In addition,  $\mathcal{P}_\tau$  ensures that no robot moves more than a distance of  $\tau/2$  towards the target points computed in  $\mathcal{P}$  and  $\mathcal{P}^{1+\tau/2}$ . The first reason is to maintain the connectivity of  $\text{UBG}(t)$ . While the protocol  $\mathcal{P}$  maintains connectivity by definition, the protocol  $\mathcal{P}^{1+\tau/2}$  could violate the connectivity of  $\text{UBG}(t)$ . Restricting the movement distance to  $\tau/2$  and upper bounding  $\tau$  by  $2/3$  resolves this issue since for all robots  $r_i$  that move according to  $\mathcal{P}^{1+\tau/2}$ ,  $\text{diam}_i(t) \leq \tau/2$ . Hence, after moving according to  $\mathcal{P}^{1+\tau/2}$ , the distance to any neighbor is at most  $3 \cdot \tau/2$ . Since  $\tau$  is upper bounded by  $2/3$ , the distance is at most 1 afterward.

► **Lemma 21.** *Let  $\mathcal{P}$  be a  $\lambda$ -contracting gathering protocol with a viewing range of 1.  $\text{UBG}(t)$  stays connected while executing  $\mathcal{P}_\tau$ .*

The second reason is that moving at most  $\tau/2$  makes sure that collisions are only possible within a range of  $\tau$ . This is crucial for our collision avoidance which is addressed in the following section. While  $\mathcal{P}_\tau$  has a viewing range of  $1 + \tau$ , it never uses its full viewing range for computing a target point. Either, it simulates  $\mathcal{P}$  with a viewing range of 1, or  $\mathcal{P}^{1+\tau/2}$  with one of  $1 + \tau/2$ . It is observable that the  $1 + \tau/2$  surrounding must always have a diameter  $\geq \tau/2$  (see Section 4.5 for more details). Hence, the diameter of robots used for the simulation of  $\mathcal{P}$  or  $\mathcal{P}^{1+\tau/2}$  cannot be less than  $\Omega(\tau)$ . The constant  $\lambda$  can be chosen accordingly.

► **Lemma 22.** *Let  $\mathcal{P}$  be a  $\lambda$ -contracting gathering protocol.  $\mathcal{P}_\tau$  is a  $\lambda'$ -contracting gathering protocol with  $\lambda' = \lambda \cdot \frac{\tau}{4(1+\tau)}$ .*

To conclude, the protocol  $\mathcal{P}_\tau$  has two main properties: it restricts the movement distance of any robot to at most  $\tau/2$  and robots  $r_i$  with  $\text{diam}_i(t) \leq \tau/2$  compute their target points based on protocol  $\mathcal{P}^{1+\tau/2}$  with viewing range  $1 + \tau/2$ .

## 4.4 Collision Avoidance

Next, we argue how to transform the protocol  $\mathcal{P}_\tau$  into the collision-free protocol  $\mathcal{P}^{cl}$ . The viewing range of  $1 + \tau$  in  $\mathcal{P}^{cl}$  allows a robot  $r_i$  to compute  $\text{target}_k^{\mathcal{P}_\tau}(t)$  (the target point in protocol  $\mathcal{P}_\tau$ ) for all robots  $r_k$  within distance at most  $\tau$ . Since the maximum movement distance of a robot in  $\mathcal{P}_\tau$  is  $\tau/2$ , this enables  $r_i$  to know the movement directions of all robots  $r_k$  which can collide with  $r_i$ . We will ensure that each robot  $r_i$  moves to some position on  $\text{colvec}_i^{\mathcal{P}_\tau}(t)$  and avoids positions of all other  $\text{colvec}_k^{\mathcal{P}_\tau}(t)$ . Henceforth, no collision can happen. While this is the basic idea of our collision avoidance, there are some details to add.

First of all,  $\mathcal{P}_\tau$  has the same viewing range as  $\mathcal{P}^{cl}$  of  $1 + \tau$ . However, it never uses the full viewing range to compute the target position  $\text{target}_i^{\mathcal{P}_\tau}(t)$ . We consider two robots  $r_i$  and  $r_k$  with distance  $\leq \tau$ . If  $r_k$  simulates  $\mathcal{P}$  to compute  $\text{target}_k^{\mathcal{P}_\tau}(t)$ ,  $r_i$  can compute  $\text{target}_k^{\mathcal{P}_\tau}(t)$  as well since  $r_i$  is able to observe all robots in distance 1 around  $r_k$ . If  $r_k$  simulates  $\mathcal{P}^{1+\tau/2}$ , the condition in  $\mathcal{P}_\tau$  makes sure that  $r_i$  and  $r_k$  have a distance of  $\leq \tau/2$ . Similarly,  $r_i$  is able to observe all robot in distance  $1 + \tau/2$  around  $r_k$  and can compute  $\text{target}_k^{\mathcal{P}_\tau}(t)$  as well.

► **Lemma 23.** *Let  $\mathcal{P}$  be a  $\lambda$ -contracting gathering protocol with a viewing range of 1. A viewing range of  $1 + \tau$  is sufficient to compute  $\text{target}_k^{\mathcal{P}_\tau}(t)$  for all robots  $r_k$  within a radius of  $\tau$ .*

## 15:20 Efficient (Near)-Gathering of Disoriented Robots with Limited Visibility

Secondly,  $r_i$  cannot avoid positions on all other  $\text{colvec}_k^{\mathcal{P}_\tau}(t)$  in some cases. For instance,  $\text{colvec}_i^{\mathcal{P}_\tau}(t)$  may be completely contained in  $\text{colvec}_k^{\mathcal{P}_\tau}(t)$  (e.g.,  $\text{colvec}_2^{\mathcal{P}_\tau}(t) \in \text{colvec}_1^{\mathcal{P}_\tau}(t)$  in the example depicted in Figure 9). In case  $\text{colvec}_i^{\mathcal{P}_\tau}(t)$  and  $\text{colvec}_k^{\mathcal{P}_\tau}(t)$  are not collinear and intersect in a single point, both robots simply avoid the intersection point (e.g.  $r_1$  and  $r_4$  in the example).

► **Lemma 24.** *No robot moves to a point that is the intersection of two collision vectors that are not collinear.*

If  $\text{colvec}_i^{\mathcal{P}_\tau}(t)$  and  $\text{colvec}_k^{\mathcal{P}_\tau}(t)$  are collinear, both robots move to a point closer to their target point than to the other one (e.g.,  $r_1$  and  $r_3$  in the example).

► **Lemma 25.** *If the target points of robots are different in  $\mathcal{P}_\tau$  they are different in  $\mathcal{P}^{cl}$ .*

But there are cases, in which robots have the same target point in  $\mathcal{P}_\tau$  (e.g.  $r_1, r_2$  and  $r_6$  in the example). Because robots stay in the same direction towards the target point, collisions can only happen if one robot is currently on the collision vector of another one (e.g.,  $r_2$  is on  $\text{colvec}_1^{\mathcal{P}_\tau}(t)$ ). Their movement is scaled by the distance to the target point, which must be different. Therefore, their target points in  $\mathcal{P}^{cl}$  must be different as well.

► **Lemma 26.** *If the target points of robots are the same in  $\mathcal{P}_\tau$  they are different in  $\mathcal{P}^{cl}$ .*

In  $\mathcal{SSYNC}$  robots may be inactive in one round. Nevertheless, in the same way, single intersection points between collision vectors and the positions of other robots are avoided as well.

► **Lemma 27.** *No robot moves to the position of an inactive robot.*

The following lemma follows immediately from Lemma 25, 26 and 27.

► **Lemma 28.** *The protocol  $\mathcal{P}^{cl}$  is collision-free.*

### 4.5 Time Bound

Previously, we have addressed the intermediate protocol  $\mathcal{P}_\tau$  that is  $\lambda$ -contracting gathering protocol concerning the viewing range of  $1 + \tau$  and also keeps  $\text{UBG}(t)$  always connected. The same holds for  $\mathcal{P}^{cl}$ . Keeping  $\text{UBG}(t)$  connected is important for the termination of a NEAR-GATHERING protocol. Suppose that  $\text{UBG}(t)$  is connected and the robots only have a viewing range of 1. Then, the robots can never decide if they can see all the other robots. However, with a viewing range of  $1 + \tau$ , it becomes possible if the swarm is brought close together ( $\text{diam}(t) < \tau$ ). For any configuration where the viewing range is  $1 + \tau$  and  $\text{UBG}(t)$  is connected, we state an important observation.

► **Lemma 29.** *Let  $\mathcal{P}$  be a  $\lambda$ -contracting protocol with viewing range  $1 + \tau$  for a constant  $\tau > 0$  and let  $\text{UBG}(t)$  be connected. If  $\text{diam}(t) > \tau$ , then  $\text{diam}_i(t) > \tau$ , for every robot  $r_i$ .*

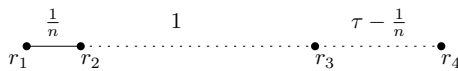
Due to the  $\lambda$ -contracting property, robots close to the boundary of the global smallest enclosing hypersphere (SEH) move upon activation at least  $\Omega\left(\frac{\text{diam}_i(t)}{\Delta}\right)$  inwards. With  $\text{diam}_i(t) > \tau$ , it follows that the radius of the SEH decreases by  $\Omega(\tau/\Delta)$  after each robot was active at least once (see Lemma 30). Consequently,  $\text{diam}(t) \leq \tau$  after  $\mathcal{O}(\Delta^2)$  epochs.

► **Lemma 30.** *Let  $\mathcal{P}$  be a  $\lambda$ -contracting protocol with a viewing range of  $1 + \tau$  while  $\text{UBG}(t)$  is always connected. After at most  $\frac{32 \cdot \pi \cdot \Delta^2}{\lambda^2 \cdot \tau} \in \mathcal{O}(\Delta^2)$  epochs executing  $\mathcal{P}$ ,  $\text{diam}(t) \leq \tau$ .*

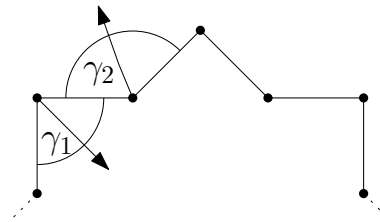
Because  $\mathcal{P}^{cl}$  has, regarding  $\lambda$ -contracting, connectivity and connectivity range, the same properties as  $\mathcal{P}_\tau$ , this lemma can directly be applied to show the runtime of Theorem 20.

## 5 Conclusion & Future Work

In this work, we introduced the class of  $\lambda$ -contracting protocols and their collision-free extensions that solve GATHERING and NEAR-GATHERING of  $n$  robots located in  $\mathbb{R}^d$  in  $\Theta(\Delta^2)$  epochs. While these results already provide several improvements over previous work, there are open questions that could be addressed by future research. First of all, we did not aim to optimize the constants in the runtime. Thus, the upper runtime bound of  $\frac{256 \cdot \pi \cdot \Delta^2}{\lambda^3}$  seems to be improvable. Moreover, one major open question remains unanswered: is it possible to solve GATHERING or NEAR-GATHERING of oblivious and disoriented robots with limited visibility in  $\mathcal{O}(\Delta)$  rounds? We could get closer to the answer: If there is such a protocol, it must compute target points regularly outside of the convex hulls of robots' neighborhoods. All  $\lambda$ -contracting protocols are slow in the configuration where the positions of the robots form a regular polygon with side length equal to the viewing range. In [15], it has been shown that this configuration can be gathered in time  $\mathcal{O}(\Delta)$  by a protocol where each robot moves as far as possible along the angle bisector between its neighbors (leaving the local convex hull). However, this protocol cannot perform well in general. See Figure 11 for the *alternating star*, a configuration where this protocol is always worse compared to any protocol that computes target points inside of local convex hulls. Figure 11 gives a hint that every protocol that performs well for the regular polygon cannot perform equally well in the alternating star. Thus, we conjecture that  $\Omega(\Delta^2)$  is a lower bound for every protocol that considers oblivious and disoriented robots with limited visibility.



■ **Figure 10** Example where target  $P_i^P(t)$  is not  $\lambda$ -centered with respect to the viewing range  $1 + \tau$ .



■ **Figure 11** The robots at  $\gamma_1$  observe a regular square, the robots at  $\gamma_2$  a regular octagon. Given that each robot moves along the angle bisector between its neighbors and leaves its local convex hull, the radius of the global SEC decreases slower than in any  $\lambda$ -contracting protocol.

## References

- 1 Sebastian Abshoff, Andreas Cord-Landwehr, Matthias Fischer, Daniel Jung, and Friedhelm Meyer auf der Heide. Gathering a closed chain of robots on a grid. In *2016 IEEE International Parallel and Distributed Processing Symposium, IPDPS 2016, Chicago, IL, USA, May 23-27, 2016*, pages 689–699. IEEE Computer Society, 2016. doi:10.1109/IPDPS.2016.51.
- 2 Noa Agmon and David Peleg. Fault-Tolerant Gathering Algorithms for Autonomous Mobile Robots. *SIAM Journal on Computing*, 36(1):56–82, January 2006. doi:10.1137/050645221.
- 3 Hideki Ando, Yoshinobu Oasa, Ichiro Suzuki, and Masafumi Yamashita. Distributed memoryless point convergence algorithm for mobile robots with limited visibility. *IEEE Trans. Robotics Autom.*, 15(5):818–828, 1999. doi:10.1109/70.795787.
- 4 Cédric Auger, Zohir Bouzid, Pierre Courtieu, Sébastien Tixeuil, and Xavier Urbain. Certified Impossibility Results for Byzantine-Tolerant Mobile Robots. In Teruo Higashino, Yoshiaki Katayama, Toshimitsu Masuzawa, Maria Potop-Butucaru, and Masafumi Yamashita, editors, *Stabilization, Safety, and Security of Distributed Systems*, Lecture Notes in Computer Science, pages 178–190, Cham, 2013. Springer International Publishing. doi:10.1007/978-3-319-03089-0\_13.

- 5 Subhash Bhagat, Sruti Gan Chaudhuri, and Krishnendu Mukhopadhyaya. Fault-tolerant gathering of asynchronous oblivious mobile robots under one-axis agreement. *J. Discrete Algorithms*, 36:50–62, 2016. doi:10.1016/j.jda.2015.10.005.
- 6 Subhash Bhagat, Sruti Gan Chaudhuri, and Krishnendu Mukhopadhyaya. Fault-Tolerant Gathering of Asynchronous Oblivious Mobile Robots under One-Axis Agreement. In M. Sohel Rahman and Etsuji Tomita, editors, *WALCOM: Algorithms and Computation*, Lecture Notes in Computer Science, pages 149–160, Cham, 2015. Springer International Publishing. doi:10.1007/978-3-319-15612-5\_14.
- 7 Subhash Bhagat and Krishnendu Mukhopadhyaya. Fault-tolerant Gathering of Semi-synchronous Robots. In *Proceedings of the 18th International Conference on Distributed Computing and Networking*, pages 1–10, Hyderabad India, January 2017. ACM. doi:10.1145/3007748.3007781.
- 8 Zohir Bouzid, Shantanu Das, and Sébastien Tixeuil. Gathering of Mobile Robots Tolerating Multiple Crash Faults. In *2013 IEEE 33rd International Conference on Distributed Computing Systems*, pages 337–346, July 2013. doi:10.1109/ICDCS.2013.27.
- 9 Zohir Bouzid, Maria Gradinariu Potop-Butucaru, and Sébastien Tixeuil. Byzantine Convergence in Robot Networks: The Price of Asynchrony. In Tarek Abdelzaher, Michel Raynal, and Nicola Santoro, editors, *Principles of Distributed Systems*, Lecture Notes in Computer Science, pages 54–70, Berlin, Heidelberg, 2009. Springer. doi:10.1007/978-3-642-10877-8\_7.
- 10 Zohir Bouzid, Maria Gradinariu Potop-Butucaru, and Sébastien Tixeuil. Optimal Byzantine-resilient Convergence in Unidimensional Robot Networks. *Theoretical Computer Science*, 411(34-36):3154–3168, July 2010. doi:10.1016/j.tcs.2010.05.006.
- 11 Quentin Bramas and Sébastien Tixeuil. Wait-Free Gathering Without Chirality. In Christian Scheideler, editor, *Structural Information and Communication Complexity*, Lecture Notes in Computer Science, pages 313–327, Cham, 2015. Springer International Publishing. doi:10.1007/978-3-319-25258-2\_22.
- 12 Philipp Brandes, Bastian Degener, Barbara Kempkes, and Friedhelm Meyer auf der Heide. Energy-efficient strategies for building short chains of mobile robots locally. *Theor. Comput. Sci.*, 509:97–112, 2013. doi:10.1016/j.tcs.2012.10.056.
- 13 Michael Braun, Jannik Castenow, and Friedhelm Meyer auf der Heide. Local gathering of mobile robots in three dimensions. In Andrea Werneck Richa and Christian Scheideler, editors, *Structural Information and Communication Complexity - 27th International Colloquium, SIROCCO 2020, Paderborn, Germany, June 29 - July 1, 2020, Proceedings*, volume 12156 of *Lecture Notes in Computer Science*, pages 63–79. Springer, 2020. doi:10.1007/978-3-030-54921-3\_4.
- 14 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. *CoRR*, abs/2206.07567, 2022.
- 15 Jannik Castenow, Jonas Harbig, Daniel Jung, Till Knollmann, and Friedhelm Meyer auf der Heide. Gathering a euclidean closed chain of robots in linear time. In Leszek Gasieniec, Ralf Klasing, and Tomasz Radzik, editors, *Algorithms for Sensor Systems - 17th International Symposium on Algorithms and Experiments for Wireless Sensor Networks, ALGOSENSORS 2021, Lisbon, Portugal, September 9-10, 2021, Proceedings*, volume 12961 of *Lecture Notes in Computer Science*, pages 29–44. Springer, 2021. doi:10.1007/978-3-030-89240-1\_3.
- 16 Mark Cieliebak, Paola Flocchini, Giuseppe Prencipe, and Nicola Santoro. Distributed Computing by Mobile Robots: Gathering. *SIAM J. Comput.*, 41(4):829–879, 2012. doi:10.1137/100796534.
- 17 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.
- 18 Reuven Cohen and David Peleg. Convergence of Autonomous Mobile Robots with Inaccurate Sensors and Movements. *SIAM J. Comput.*, 38(1):276–302, 2008. doi:10.1137/060665257.

- 19 Andreas Cord-Landwehr, Bastian Degener, Matthias Fischer, Martina Hüllmann, Barbara Kempkes, Alexander Klaas, Peter Kling, Sven Kurras, Marcus Märten, Friedhelm Meyer auf der Heide, Christoph Raupach, Kamil Swierkot, Daniel Warner, Christoph Weddemann, and Daniel Wonisch. A New Approach for Analyzing Convergence Algorithms for Mobile Robots. In Luca Aceto, Monika Henzinger, and Jirí Sgall, editors, *Automata, Languages and Programming*, Lecture Notes in Computer Science, pages 650–661, Berlin, Heidelberg, 2011. Springer. doi:10.1007/978-3-642-22012-8\_52.
- 20 Andreas Cord-Landwehr, Matthias Fischer, Daniel Jung, and Friedhelm Meyer auf der Heide. Asymptotically optimal gathering on a grid. In Christian Scheideler and Seth Gilbert, editors, *Proceedings of the 28th ACM Symposium on Parallelism in Algorithms and Architectures, SPAA 2016, Asilomar State Beach/Pacific Grove, CA, USA, July 11-13, 2016*, pages 301–312. ACM, 2016. doi:10.1145/2935764.2935789.
- 21 Shantanu Das, Paola Flocchini, Nicola Santoro, and Masafumi Yamashita. On the computational power of oblivious robots: forming a series of geometric patterns. In Andréa W. Richa and Rachid Guerraoui, editors, *Proceedings of the 29th Annual ACM Symposium on Principles of Distributed Computing, PODC 2010, Zurich, Switzerland, July 25-28, 2010*, pages 267–276. ACM, 2010. doi:10.1145/1835698.1835761.
- 22 Xavier Défago, Maria Gradinariu, Stéphane Messika, and Philippe Raipin-Parvédy. Fault-Tolerant and Self-stabilizing Mobile Robots Gathering. In Shlomi Dolev, editor, *Distributed Computing*, Lecture Notes in Computer Science, pages 46–60, Berlin, Heidelberg, 2006. Springer. doi:10.1007/11864219\_4.
- 23 Bastian Degener, Barbara Kempkes, Peter Kling, and Friedhelm Meyer auf der Heide. Linear and competitive strategies for continuous robot formation problems. *ACM Trans. Parallel Comput.*, 2(1):2:1–2:18, 2015. doi:10.1145/2742341.
- 24 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.
- 25 Yoann Dieudonné, Franck Petit, and Vincent Villain. Leader election problem versus pattern formation problem. In Nancy A. Lynch and Alexander A. Shvartsman, editors, *Distributed Computing, 24th International Symposium, DISC 2010, Cambridge, MA, USA, September 13-15, 2010. Proceedings*, volume 6343 of *Lecture Notes in Computer Science*, pages 267–281. Springer, 2010. doi:10.1007/978-3-642-15763-9\_26.
- 26 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.
- 27 Paola Flocchini, Giuseppe Prencipe, and Nicola Santoro. Moving and computing models: 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 3–14. Springer, 2019. doi:10.1007/978-3-030-11072-7\_1.
- 28 Paola Flocchini, Giuseppe Prencipe, Nicola Santoro, and Giovanni Viglietta. Distributed computing by mobile robots: uniform circle formation. *Distributed Comput.*, 30(6):413–457, 2017. doi:10.1007/s00446-016-0291-x.
- 29 Paola Flocchini, Giuseppe Prencipe, Nicola Santoro, and Peter Widmayer. Gathering of asynchronous robots with limited visibility. *Theor. Comput. Sci.*, 337(1-3):147–168, 2005. doi:10.1016/j.tcs.2005.01.001.
- 30 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. doi:10.1016/j.tcs.2008.07.026.

- 31 Taisuke Izumi, Zohir Bouzid, Sébastien Tixeuil, and Koichi Wada. The BG-simulation for Byzantine Mobile Robots, June 2011. doi:10.48550/arXiv.1106.0113.
- 32 Taisuke Izumi, Zohir Bouzid, Sébastien Tixeuil, and Koichi Wada. Brief Announcement: The BG-Simulation for Byzantine Mobile Robots. In David Peleg, editor, *Distributed Computing*, Lecture Notes in Computer Science, pages 330–331, Berlin, Heidelberg, 2011. Springer. doi:10.1007/978-3-642-24100-0\_32.
- 33 Taisuke Izumi, Samia Souissi, Yoshiaki Katayama, Nobuhiro Inuzuka, Xavier Défago, Koichi Wada, and Masafumi Yamashita. The Gathering Problem for Two Oblivious Robots with Unreliable Compasses. *SIAM J. Comput.*, 41(1):26–46, 2012. doi:10.1137/100797916.
- 34 Heinrich Jung. Ueber die kleinste kugel, die eine räumliche figur einschliesst. *Journal für die reine und angewandte Mathematik*, 123:241–257, 1901. URL: <http://eudml.org/doc/149122>.
- 35 Heinrich Jung. Über den kleinsten kreis, der eine ebene figur einschließt. *Journal für die reine und angewandte Mathematik*, 137:310–313, 1910. URL: <http://eudml.org/doc/149324>.
- 36 Branislav Katreniak. Convergence with Limited Visibility by Asynchronous Mobile Robots. In Adrian Kosowski and Masafumi Yamashita, editors, *Structural Information and Communication Complexity - 18th International Colloquium, SIROCCO 2011, Gdansk, Poland, June 26-29, 2011. Proceedings*, volume 6796 of *Lecture Notes in Computer Science*, pages 125–137. Springer, 2011. doi:10.1007/978-3-642-22212-2\_12.
- 37 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.
- 38 Shouwei Li, Christine Markarian, Friedhelm Meyer auf der Heide, and Pavel Podlipyan. A continuous strategy for collisionless gathering. *Theor. Comput. Sci.*, 852:41–60, 2021. doi:10.1016/j.tcs.2020.10.037.
- 39 Ji Lin, A. Stephen Morse, and Brian D. O. Anderson. The multi-agent rendezvous problem. part 1: The synchronous case. *SIAM J. Control. Optim.*, 46(6):2096–2119, 2007. doi:10.1137/040620552.
- 40 Ji Lin, A. Stephen Morse, and Brian D. O. Anderson. The multi-agent rendezvous problem. part 2: The asynchronous case. *SIAM J. Control. Optim.*, 46(6):2120–2147, 2007. doi:10.1137/040620564.
- 41 Linda Pagli, Giuseppe Prencipe, and Giovanni Viglietta. Getting close without touching. In Guy Even and Magnús M. Halldórsson, editors, *Structural Information and Communication Complexity - 19th International Colloquium, SIROCCO 2012, Reykjavik, Iceland, June 30-July 2, 2012, Revised Selected Papers*, volume 7355 of *Lecture Notes in Computer Science*, pages 315–326. Springer, 2012. doi:10.1007/978-3-642-31104-8\_27.
- 42 Linda Pagli, Giuseppe Prencipe, and Giovanni Viglietta. Getting close without touching: near-gathering for autonomous mobile robots. *Distributed Comput.*, 28(5):333–349, 2015. doi:10.1007/s00446-015-0248-5.
- 43 Debasish Pattanayak, Kaushik Mondal, H. Ramesh, and Partha Sarathi Mandal. Fault-Tolerant Gathering of Mobile Robots with Weak Multiplicity Detection. In *Proceedings of the 18th International Conference on Distributed Computing and Networking*, pages 1–4, Hyderabad India, January 2017. ACM. doi:10.1145/3007748.3007786.
- 44 Pavan Poudel and Gokarna Sharma. Time-optimal gathering under limited visibility with one-axis agreement. *Inf.*, 12(11):448, 2021. doi:10.3390/info12110448.
- 45 Giuseppe Prencipe. Impossibility of gathering by a set of autonomous mobile robots. *Theor. Comput. Sci.*, 384(2-3):222–231, 2007. doi:10.1016/j.tcs.2007.04.023.
- 46 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.



- 47 Masafumi Yamashita and Ichiro Suzuki. Characterizing geometric patterns formable by oblivious anonymous mobile robots. *Theor. Comput. Sci.*, 411(26-28):2433–2453, 2010. doi:10.1016/j.tcs.2010.01.037.
- 48 Yukiko Yamauchi, Taichi Uehara, and Masafumi Yamashita. Brief announcement: Pattern formation problem for synchronous mobile robots in the three dimensional euclidean space. In George Giakkoupis, editor, *Proceedings of the 2016 ACM Symposium on Principles of Distributed Computing, PODC 2016, Chicago, IL, USA, July 25-28, 2016*, pages 447–449. ACM, 2016. doi:10.1145/2933057.2933063.
- 49 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*, Lecture Notes in Computer Science, pages 201–212, Cham, 2013. Springer International Publishing. doi:10.1007/978-3-319-03578-9\_17.