Brief Announcement: Distinct Gathering Under Round Robin

Fabian Frei
CISPA Helmholtz Center for Information Security, Saarbrücken, Germany
Department of Computer Science, ETH Zürich, Switzerland

Koichi Wada
Hosei University, Tokyo, Japan

Abstract

We resolve one of the longest-standing questions about autonomous mobile robots in a surprising way.

Distinct Gathering is the fundamental cooperation task of letting robots, initially scattered across the plane in distinct locations, gather in an arbitrary single point. The scheduler *Round Robin* cyclically activates the robots one by one in a fixed order. When activated, a robot perceives all robot locations and moves wherever it wants based only on this information. For $n = 2$ robots, the task is trivial. What happens for $n \geq 3$ has remained an open problem for decades by now. The established conjecture declares the task to be impossible in this case. We prove that it is indeed impossible for $n = 3$ but, to great surprise, possible again for any $n \geq 4$. We go beyond the standard requirements by providing a very robust algorithm that does not require any consistency or self-consistency for the local Cartesian maps perceived by the robots and works even for non-rigid movement, that is, if robots may be unpredictably stopped and deactivated during a movement.

2012 ACM Subject Classification Computer systems organization → Robotic autonomy

Keywords and phrases Autonomous mobile robots, Distinct Gathering, Round Robin

Digital Object Identifier 10.4230/LIPIcs.DISC.2024.48

Funding *Koichi Wada*: Supported by JSPS KAKENHI Grants 20K11685 and 21K11748.

Acknowledgements We are very grateful to all reviewers who helped us to improve this paper.

1 Introduction

Mobile robotics is an active field that studies which cooperation tasks can be performed by simplistic mobile robots. The underlying motivation is to develop cheap, reliable, and robust robots to be employed in disaster relief, for example. Suzuki and Yamashita [25] introduced the default Look-Compute-Move model: The robots are all identical, oblivious, and represented by points in the Euclidean plane. Whenever activated by some *scheduler*, a robot observes all robot locations in a local Cartesian coordinate system (which might be arbitrarily scaled, rotated, and mirrored with respect to the global coordinate system) and then moves wherever it wants. This model is sometimes referred to as OBLOT (for *oblivious robots*), and has been the object of intensive study [1, 2, 3, 4, 5, 6, 11, 12, 15, 16, 18, 23]. The simplest scheduler is the *fully synchronous* one, which always activates all robots synchronously. The second most basic and simple scheduler is arguably *Round Robin*, which lets the all robots move in orderly turns, one by one, always in the same order. The standard textbook by Flocchini et al. [13] and the newer textbook [10] provide an excellent overview of the basic models and literature.

Gathering all robots in a single point is arguably the most fundamental cooperation task and has consequently garnered the attention of many researchers [1, 2, 3, 4, 5, 6, 7, 8, 11, 14, 17, 18, 19, 20, 22, 24]. Gathering is trivial under the fully synchronous scheduler; each robot moves to the center of gravity, where they meet after a single activation. This strategy fails for Round Robin; the robots will converge but never meet. Instead, the easy solution is to target a point of maximum multiplicity [25, Thm. 3.4]. Without multiplicity detection, gathering more than two robots is much more difficult. Indeed, it becomes impossible, at least if we weaken the model such to allow for initial configurations where some robots occupy the same location [21, Sec. 3]. This task variant is sometimes called *Self-Stabilizing Gathering* to distinguish it from the original task introduced in Suzuki and Yamashita's seminal paper [25], which assumes that the robots all start from distinct locations. This original version used to be called just Gathering, but is nowadays often referred to as *Distinct* Gathering to avoid ambiguity. It is an obvious and by now decades-old research problem whether Distinct Gathering is still possible for more than two robots under Round Robin, the simplest scheduler besides the fully synchronous one. Most recently, it was tackled again by an algorithm that solves the problem at least under the two additional assumptions that the robots have some internal memory and share a unit distance [26, Sect. 4, Alg. 10]. But the general consensus remained that Distinct Gathering is likely impossible under Round Robin without additional assumptions. A conjecture by Défago et al. made this explicit. [9, Conj. 2]. They corroborated their conjecture by proving the impossibility under the plausible assumption that any successful algorithm is a so-called *rapid-gathering* algorithm [9, Conj. 1]. Our results disprove both of these conjectures and resolve the open question by showing that Distinct Gathering under Round Robin, while indeed impossible for three robots, unexpectedly becomes feasible again for four and more robots. We go beyond the default robustness requirements by considering *non-rigid movement with an unknown minimum movement distance*: Any robot may be stopped at arbitrary times after some unknown minimal movement distance (e.g., due to overheating). This causes it to lose all of its memory and leaves it to wait for its next activation by the scheduler. We can provide an algorithm that works even under these exceptionally adverse conditions.

2 Detailed Model Description

In this section, we define the task to be solved, the robot model, and the considered scheduler.

▶ **Definition 1** (Gathering and Distinct Gathering). Gathering *is the task to gather n robots, which are initially scattered arbitrarily across the Euclidean plane in unknown locations. The gathering is achieved if after a finite number of activation steps all robots stay co-located in a single point forever – which point of the plane the robots choose as their gathering point is irrelevant.* Distinct gathering *is the same task with the additional guarantee that all robots occupy different locations in the initial configuration. We call a configuration with this property that no two robots share a location, a* distinct *configuration.*

▶ **Definition 2** (Robot Model). *The robots are all* identical *and* anonymous *(so they are indistinguishable and do not even have any internal ID) and all run the same algorithm. They are very simplistic and can see and move only when activated by an external scheduler. Upon activation a robot always executes a so-called Look-Compute-Move cycle:*

Look-Compute-Move Cycle. *The robot first looks around and detects all current robot locations. Then it uses this information, which is sometimes called a* snapshot*, to decide where to move and what trajectory to follow. For the upper bounds announced in this paper, we even restrict ourselves to robots moving on straight lines towards the computed target. Finally, the robot moves to its target along the pre-computed path, ignoring everything else.*

In particular, the robots are crash-resistant*; there movement is not affected in any way by them passing each other or being collocated with each other. Under the often standard assumption of* rigid *movement, the robot always reaches its target, which concludes the Look-Compute-Move cycle.*

Local Maps. *Our robots are located in the Euclidean plane and assumed to be dimensionless; they can thus be represented by points. The robots perceive the locations of the other robots with absolute precision, but only relative to a* Cartesian map *(i.e., a map with two orthogonal axes that have the same unit length) with the observing robot at the origin. Importantly, there are no further guarantees for consistency between the robots, neither for the perceived unit length nor the rotation of the axes nor the chirality. (Gathering would indeed be trivial with fully consistent maps.) Moreover, we do not even assume any self-consistency for these properties either. In other words, an adversary may arbitrarily scale, rotate, and mirror the map perceived by the robot upon each activation.*

No Multiplicity Detection. *We assume that the robots cannot detect multiplicity, that is, whether any perceived location is shared by multiple robots or not. This includes the location of the observing robot itself; it cannot sense directly whether or not it is accompanied.*

Obliviousness. *The robots are* stateless *(often referred to as* oblivious*): They possess no persistent memory and can thus only use the location information gathered upon activation to compute their movement path, and as soon as a robot stops, it forgets this information, too.*

We now define the scheduler under which Distinct Gathering is examined in this paper.

▶ **Definition 3** (Round Robin)**.** Round Robin *is the scheduler the activates the robots one by one, according to a fixed but previously unknown order, in an ever-continuing cycle covering all robots. We call any consecutive activation of all n robots a* round*. Just as the fully-synchronous scheduler, Round Robin is a natural special case of the* semi-synchronous *scheduler; that is, it always waits with the activation of a robot until all robots have stopped moving.*

Finally, we distinguish between the standard model of *rigid movement* and the assumption of the less reliable *non-rigid movement*, which models various adverse situations such as robots overheating or running out of energy during their movement.

▶ **Definition 4** (Rigid and Non-Rigid Movement)**.** *Robots with so-called* rigid movement *always reach their target when executing a Look-Compute-Move. The opposite is* non-rigid movement*, where a robot's movement may be stopped at any time by an adversary, as long as the robot has moved some minimum distance during its current cycle. This prematurely ends the robot's Look-Compute-Move cycle. For our algorithm, we assume the harshest variant of non-rigid movement where the robots have no knowledge of the minimum movement distances.*

3 Previous Results and Conjectures

Distinct Gathering is trivial under Round Robin for the case of $n = 2$ robots, often referred to as Rendezvous: It suffices for every activated robot to target to location of the other robot. The task becomes far more interesting and challenging for $n \geq 3$, however. Whether it is feasible or not has remained a prominent open question in the field for over two decades by now. It has been tackled repeatedly, most recently in a paper by Terai et al., who showed how to solve the problem at least using the quite strong additional assumptions that the robots have some internal memory, share a unit distance, and that all movements are rigid [26, Sect. 4, Alg. 10]. But the general consensus has remained that Distinct Gathering is most likely impossible under Round Robin under the standard assumptions outlined above. The following conjecture by Défago et al. makes this explicit. [9, Conj. 2].

▶ **Conjecture 5** (Cf. Défago et al. [9, Conj. 2]). *There is no algorithm solving Distinct Gathering under Round Robin for $n \geq 3$ robots under the default assumptions detailed in Section 2.*

They corroborate their assumption by proving [9, Thm. 16] the impossibility of Distinct Gathering under Round Robin under the assumption that any algorithm solving this problem is a so-called *rapid-gathering* algorithm [9, Conj. 1].

▶ **Definition 6** (Rapid-gathering algorithm, cf. Défago et al. [9, Def. 1]). *Any algorithm is a* rapid-gathering *algorithm if there is an initial configuration of n robots in distinct locations and a Round Robin activation order such that the algorithm gathers all robots within $n-1$ activations.*

▶ **Conjecture 7** (Cf. Défago et al. [9, Conj. 1]). *Every algorithm for Gathering under Round Robin is a rapid-gathering algorithm.*

They further substantiate this plausible claim by considering a series of convincing examples.

4 New Results

We are able to resolve the open question about the feasibility of Distinct Gathering under Round Robin, giving a quite surprising answer that might seem contradictory at first sight.

On the one hand, we prove with Theorem 8 that the problem is indeed impossible to solve for the case of $n = 3$. On the other hand, we show with Theorem 9 that it unexpectedly becomes feasible again for four and more robots, even though these robots have to pass through three-point configurations during the execution of their algorithm. We first formally state the result for three robots.

▶ **Theorem 8.** *Distinct Gathering under the Round Robin is impossible for $n = 3$ robots under the default model detailed in Section 2.*

Proof Sketch. Due to the space constraints, we can unfortunately not even provide a full proof sketch here since some of the details are quite subtle and require an extensive description. We restrict ourselves to mentioning here the following. The proof relies on first assuming towards contradiction that there is a gathering algorithm, then choosing an arbitrary initial configuration and a schedule, then considering the last configuration before last transition from a three-point configuration to a two-point configuration by the given algorithm under this schedule, and then considering both what could or could not have happened with this mentioned configuration as an alternative initial configuration under the different possible schedules and how the algorithm must further behave under the originally chosen schedule. ◀

We remark that Theorem 8 unifies and strengthens two separate impossibility results by Défago et al. [9, Theorems 14 and 15] at the same time for $n = 3$, extending one from (Self-Stabilizing) Gathering to *Distinct* Gathering and the other from so-called k-bounded schedulers to the heavily restricted *Round Robin* scheduler[1]. It confirms both Conjecture 5 and Conjecture 7 in the case $n = 3$. The natural assumption would now be that this result extends beyond three robots to the case of $n \geq 4$ because gathering more robots seems to only increase the difficulty of the problem. Indeed, any initial distinct configuration with four or more robots is forced to pass through a three-point configuration under Round Robin before a gathering can be achieved. Surprisingly, we can show that this argumentation is flawed by providing an algorithm that solves the problems for any given $n \geq 4$.

▶ **Theorem 9.** *Distinct Gathering under Round Robin can be solved for any given $n \geq 4$ under the default model detailed in Section 2. The result holds even for robots restricted to moving to their target in a straight line and assuming non-rigid movement with an unknown minimal movement distance.*

Theorem 9 disproves the established Conjectures 5 and 7 and resolves the long-standing open question about the feasibility of Distinct Gathering under Round Robin.

Proof Sketch. It is possible to distinguish seven main phases of the algorithm. It typically passes through these phases in the given order, but some phases may be skipped depending on the initial configuration. Moreover, in some cases it becomes necessary for the algorithm to restart the current phase or even return to previous phases. A graphic representation of the phases and the possible transitions between them is given in Figure 1.

Due to the space restrictions we can only provide a brief outline of the purpose of each phase.

The purpose of the first phase is to create a configuration whose robot locations form at most one *isosceles right triangle* or multiple isosceles right triangles with one robot that is part of all of them. The first phase ends once such a configuration is attained. The second phase has a very similar goal, namely a configuration with *exactly one* isosceles right triangle or multiple isosceles right triangles with one robot that is part of all of them. The third phase creates a configuration with exactly one isosceles right triangle. Everything up to start of the fourth phase maintains distinct locations for all the robots. In the fourth phase, all robots except for the one located at the vertex of the isosceles right triangle move to the closer one of the two endpoints of the base of this triangle. In the fifth phase, the robot at the vertex moves down along the altitude to the midpoint of the triangle's base, resulting in a *midpoint configuration*, which means that one robot is exactly at the midpoint between two other ones. In the sixth phase, all robots start targeting this midpoint. This phase ends as soon as a two-point configuration is created or a robot fails to reach its target. In the latter case, a non-midpoint three-point configuration may result, which triggers the start of the eighth phase that restores a midpoint configuration. If a four-point configuration results at the end of the sixth phase, then the seventh phase begins and ends by creating another unique isosceles right triangle, leading back to the fourth phase. The last option at the end of the sixth phase is that a two-point configuration is created, which starts the ninth phase. In this phase, all robots target the opposite observed location. This can either end in a gathering or create another three-point configuration, which means moving back to the sixth or eighth phase. One might suspect that the algorithm could be stuck in an infinite

[1] Note that the cited paper implicitly assumes neither consistency nor self-consistency in the maps of the robots, not even for the unit distance, as evidenced by the remark in the proof of Theorem 14 that all two-point configurations are indistinguishable. For the cited proofs, this assumption is insubstantial and easily removed by considering not arbitrary, but a very specific two-point configurations. In other cases, however, it might be crucial whether the perceived unit distance is self-consistent for each robot.

loop of robots swapping locations in the ninth phase. The proof that this does not happen relies an a sequence of invariants maintained through the execution of the previous phases. To prove termination of the algorithm, we need to show the termination of each phase, but also that looping back to previous phases can happen only a finite number of times. This is possible by showing that the configuration's diameter decreases substantially with every throwback to a previous phase, which lets us assume rigid movement after a finite number of phase transitions. Rigid movement prevents any transition back to the sixth, seventh, or eighth phase, and thus guarantees that the robots eventually all gather at the end of the ninth phase. ◀

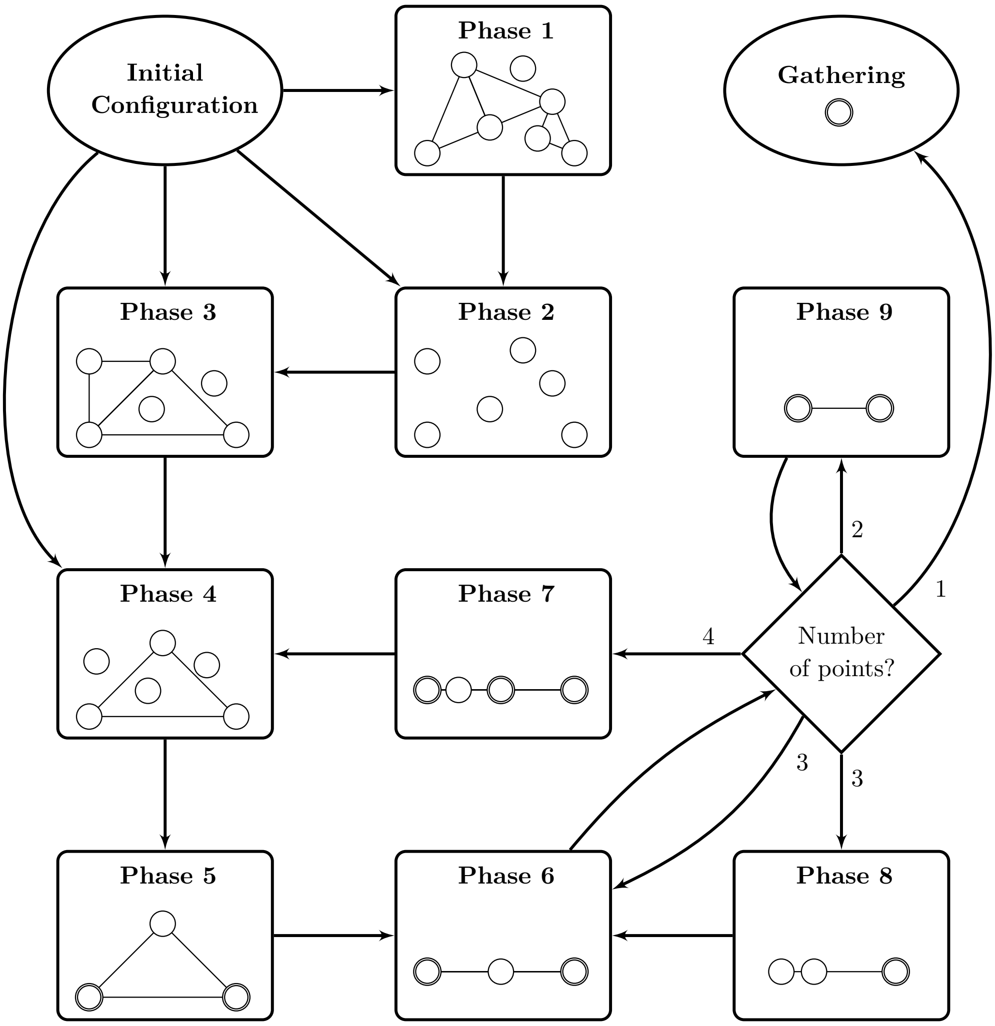

Figure 1 The phase transition diagram for the algorithm solving Distinct Gathering for more than three robots under Round Robin. Typical configurations that might occur at the start of each phase are shown. The arrows indicate the possible phase transitions. It is proved that no infinite loops can occur.

References

1 N. Agmon and D. Peleg. Fault-tolerant gathering algorithms for autonomous mobile robots. *SIAM Journal on Computing*, 36(1):56–82, 2006. doi:10.1137/050645221.

2 H. Ando, Y. Osawa, I. Suzuki, and M. Yamashita. A distributed memoryless point convergence algorithm for mobile robots with limited visivility. *IEEE Transactions on Robotics and Automation*, 15(5):818–828, 1999. doi:10.1109/70.795787.

3 Z. Bouzid, S. Das, and S. Tixeuil. Gathering of mobile robots tolerating multiple crash faults. In *the 33rd Int. Conf. on Distributed Computing Systems*, pages 334–346, 2013.

4 S. Cicerone, Di Stefano, and A. Navarra. Gathering of robots on meeting-points. *Distributed Computing*, 31(1):1–50, 2018.

5 M. Cieliebak, P. Flocchini, G. Prencipe, and N. Santoro. Distributed computing by mobile robots: Gathering. *SIAM Journal on Computing*, 41(4):829–879, 2012. doi:10.1137/100796534.

6 R. Cohen and D. Peleg. Convergence properties of the gravitational algorithms in asynchronous robot systems. *SIAM J. on Computing*, 34(15):1516–1528, 2005. doi:10.1137/S0097539704446475.

7 S. Das, P. Flocchini, G. Prencipe, N. Santoro, and M. Yamashita. Autonomous mobile robots with lights. *Theoretical Computer Science*, 609:171–184, 2016. doi:10.1016/J.TCS.2015.09.018.

8 X. Défago, M. Potop-Butucaru, and Philippe Raipin-Parvédy. Self-stabilizing gathering of mobile robots under crash or byzantine faults. *Distributed Computing*, 33(5):393–421, 2020. doi:10.1007/S00446-019-00359-X.

9 Xavier Défago, Maria Potop-Butucaru, and Philippe Raipin Parvédy. Self-stabilizing gathering of mobile robots under crash or byzantine faults. *Distributed Comput.*, 33(5):393–421, 2020. doi:10.1007/s00446-019-00359-x.

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

11 P. Flocchini, G. Prencipe, N. Santoro, and P. Widmayer. Gathering of asynchronous robots with limited visibility. *Theoretical Computer Science*, 337(1–3):147–169, 2005. doi:10.1016/J.TCS.2005.01.001.

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

13 Paola Flocchini, Giuseppe Prencipe, and Nicola Santoro. *Distributed Computing by Oblivious Mobile Robots*. Synthesis Lectures on Distributed Computing Theory. Morgan and Claypool Publishers, 2012.

14 Paola Flocchini, Nicola Santoro, and Koichi Wada. On memory, communication, and synchronous schedulers when moving and computing. In *Proceeding of the 23rd International Conference on Principles of Distributed Systems (OPODIS 2019)*, volume 153 of *LIPIcs*, pages 25:1–25:17, 2019. doi:10.4230/LIPICS.OPODIS.2019.25.

15 N. Fujinaga, Y. Yamauchi, H. Ono, S. Kijima, and M. Yamashita. Pattern formation by oblivious asynchronous mobile robots. *SIAM Journal on Computing*, 44(3):740–785, 2015. doi:10.1137/140958682.

16 V. Gervasi and G. Prencipe. Coordination without communication: The case of the flocking problem. *Discrete Applied Mathematics*, 144(3):324–344, 2004. doi:10.1016/J.DAM.2003.11.010.

17 T. Izumi, Y. Katayama, N. Inuzuka, and K. Wada. Gathering autonomous mobile robots with dynamic compasses: An optimal result. In *21st DISC*, pages 298–312, 2007.

18 T. Izumi, S. Souissi, Y. Katayama, N. Inuzuka, X. Défago, K. Wada, and M. Yamashita. The gathering problem for two oblivious robots with unreliable compasses. *SIAM Journal on Computing*, 41(1):26–46, 2012. doi:10.1137/100797916.

19 J. Lin, A.S. Morse, and B.D.O. Anderson. The multi-agent rendezvous problem. parts 1 and 2. *SIAM Journal on Computing*, 46(6):2096–2147, 2007.

20 T. Okumura, K. Wada, and X. Défago. Optimal rendezvous $\mathcal{L}$-algorithms for asynchronous mobile robots with external-lights. *Theoretical Computer Science*, 979(114198), 2023.

21 Giuseppe Prencipe. On the feasibility of gathering by autonomous mobile robots. In Andrzej Pelc and Michel Raynal, editors, *Proceedings of the 12th International Colloquium on Structural Information and Communication Complexity (SIROCCO 2005)*, volume 3499 of *Lecture Notes in Computer Science*, pages 246–261. Springer, 2005. doi:10.1007/11429647_20.

22 S. Souissi, X. Défago, and M. Yamashita. Using eventually consistent compasses to gather memory-less mobile robots with limited visibility. *ACM Transactions on Autonomous and Adaptive Systems*, 4(1):1–27, 2009. doi:10.1145/1462187.1462196.

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

24 I. Suzuki and M. Yamashita. Distributed anonymous mobile robots: Formation of geometric patterns. *SIAM Journal on Computing*, 28:1347–1363, 1999. doi:10.1137/S009753979628292X.

25 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.

26 Satoshi Terai, Koichi Wada, and Yoshiaki Katayama. Gathering problems for autonomous mobile robots with lights. *Theor. Comput. Sci.*, 941:241–261, 2023. doi:10.1016/j.tcs.2022.11.018.