Deadlock and Noise in Self-Organized Aggregation Without Computation

Joshua J. Daymude[1(✉)], Noble C. Harasha[2], Andréa W. Richa[3], and Ryan Yiu[3]

[1] Biodesign Center for Biocomputing, Security and Society, Arizona State University, Tempe, AZ 85281, USA
jdaymude@asu.edu

[2] Massachusetts Institute of Technology, Cambridge, MA 02139, USA
nharasha@mit.edu

[3] School of Computing and Augmented Intelligence, Arizona State University, Tempe, AZ 85281, USA
aricha@asu.edu

Abstract. Aggregation is a fundamental behavior for swarm robotics that requires a system to gather together in a compact, connected cluster. In 2014, Gauci et al. proposed a surprising algorithm that reliably achieves swarm aggregation using only a binary line-of-sight sensor and no arithmetic computation or persistent memory. It has been rigorously proven that this algorithm will aggregate one robot to another, but it remained open whether it would always aggregate a system of $n > 2$ robots as was observed in experiments and simulations. We prove that there exist deadlocked configurations from which this algorithm cannot achieve aggregation for $n > 3$ robots when the robots' motion is uniform and deterministic. In practice, however, the physics of collisions and slipping work to the algorithm's advantage in avoiding deadlock; moreover, we show that the algorithm is robust to small amounts of noise in its sensors and in its motion. Finally, we prove that the algorithm achieves a linear runtime speedup for the $n = 2$ case when using a cone-of-sight sensor instead of a line-of-sight sensor.

Keywords: Swarm robotics · Self-organization · Aggregation

1 Introduction

The fields of swarm robotics [5,14,15,24] and programmable matter [2,18,33] seek to engineer systems of simple, easily manufactured robot modules that can cooperate to perform tasks involving collective movement and reconfiguration. Our present focus is on the *aggregation problem* (also referred to as

The authors gratefully acknowledge support from the U.S. ARO under MURI award #W911NF-19-1-0233 and from the Arizona State University Biodesign Institute.

C. Johnen et al. (Eds.): SSS 2021, LNCS 13046, pp. 51–65, 2021.
https://doi.org/10.1007/978-3-030-91081-5_4

"gathering" [8,16,19] and "rendezvous" [9,34,35]) in which a robot swarm must gather together in a compact, connected cluster [3]. Aggregation has a rich history in swarm robotics as a prerequisite for other collective behaviors requiring densely connected swarms. Inspired by self-organizing aggregation in nature [6,12,13,25,27,29], numerous approaches for swarm aggregation have been proposed, each one seeking to achieve aggregation faster, more robustly, and with less capable individuals than the last [1,11,17,26,28].

One goal from the theoretical perspective has been to identify *minimal capabilities* for an individual robot such that a collective can provably accomplish a given task. Towards this goal, Roderich Groß and others at the Natural Robotics Laboratory have developed a series of very simple algorithms for swarm behaviors like spatially sorting by size [7,23], aggregation [21], consensus [32], and coverage [31]. These algorithms use at most a few bits of sensory information and express their entire structure as a single "if-then-else" statement, avoiding any arithmetic computation or persistent memory. Although these algorithms have been shown to perform well in both robotic experiments and simulations with larger swarms, some lack general, rigorous proofs that guarantee the correctness of the swarm's behavior.

In this work, we investigate the swarm aggregation algorithm of Gauci et al. [21] (summarized in Sect. 2) whose provable convergence for systems of $n > 2$ robots remained an open question. In Sect. 3, we answer this question negatively, identifying deadlocked configurations from which aggregation is never achieved. Motivated by the need to break these deadlocks, we corroborate and extend the simulation results of [20] by showing that the algorithm is robust to two distinct forms of error (Sect. 4). Finally, we prove that the time required for a single robot to aggregate to a static robot improves by a linear factor when using a cone-of-sight sensor instead of a line-of-sight sensor; however, simulations show this comparative advantage decreases for larger swarms (Sect. 5).

2 The Gauci et al. Swarm Aggregation Algorithm

Given n robots in arbitrary initial positions on the two-dimensional plane, the goal of the *aggregation problem* is to define a controller that, when used by each robot in the swarm, eventually forms a compact, connected cluster. Gauci et al. [21] introduced an algorithm for aggregation among e-puck robots [30] that only requires binary information from a robot's (infinite range) line-of-sight sensor indicating whether it sees another robot ($I = 1$) or not ($I = 0$). The controller $x = (v_{\ell 0}, v_{r0}, v_{\ell 1}, v_{r1}) \in [-1, 1]^4$ actuates the left and right wheels according to velocities $(v_{\ell 0}, v_{r0})$ if $I = 0$ and $(v_{\ell 1}, v_{r1})$ otherwise. Using a grid search over a sufficiently fine-grained parameter space and evaluating aggregation according to a dispersion metric, they determined that the best controller was:

$$x^* = (-0.7, -1, 1, -1).$$

Thus, when no robot is seen, a robot using x^* will rotate around a point c that is 90° counter-clockwise from its line-of-sight sensor and $R = 14.45$ cm away at a speed of $\omega_0 = -0.75$ rad/s; when a robot is seen, it will rotate clockwise in place at a speed of $\omega_1 = -5.02$ rad/s. The following three theorems summarize the theoretical results for this aggregation algorithm.

Theorem 1 (Gauci et al. [21]). *If the line-of-sight sensor has finite range, then for every controller x there exists an initial configuration in which the robots form a connected visibility graph but from which aggregation will never occur.*

Theorem 2 (Gauci et al. [21]). *One robot using controller x^* will always aggregate to another static robot or static circular cluster of robots.*

Theorem 3 (Gauci et al. [21]). *Two robots both using controller x^* will always aggregate.*

Our main goal, then, is to investigate the following conjecture that is well-supported by evidence from simulations and experiments.

Conjecture 1. A system of $n > 2$ robots each using controller x^* will always aggregate.

Throughout the remaining sections, we measure the degree of aggregation in the system using the following metrics:

- *Smallest Enclosing Disc (SED) Circumference.* The smallest enclosing disc of a set of points S in the plane is the circular region of the plane containing S and having the smallest possible radius. Smaller circumferences correspond to more aggregated configurations.
- *Convex Hull Perimeter.* The convex hull of a set of points S in the plane is the smallest convex polygon enclosing S. Smaller perimeters correspond to more aggregated configurations. Due to the flexibility of convex polygons, this metric is less sensitive to outliers than the smallest enclosing disc which is forced to consider a circular region.
- *Dispersion (2nd Moment).* Adapting Gauci et al. [21] and Graham and Sloane [22], let p_i denote the (x, y)-coordinate of robot i on the continuous plane and $\overline{p} = \frac{1}{n}\sum_{i=1}^{n} p_i$ be the centroid of the system. Dispersion is defined as:
$$\sum_{i=1}^{n} ||p_i - \overline{p}||_2 = \sum_{i=1}^{n} \sqrt{(x_i - \overline{x})^2 + (y_i - \overline{y})^2}$$
Smaller values of dispersion correspond to more aggregated configurations.
- *Cluster Fraction.* A cluster is a set of robots that is connected by means of (nearly) touching. Following Gauci et al. [21], our final metric for aggregation is the fraction of robots in the largest cluster. Unlike the previous metrics, larger cluster fractions correspond to more aggregated configurations.

We use dispersion as our primary metric of aggregation since it is the metric that is least sensitive to outliers and was used by Gauci et al. [21], enabling a clear comparison of results.

3 Impossibility of Aggregation for $n > 3$ Robots

In this section, we rigorously establish a negative result indicating that Conjecture 1 does not hold in general. This result identifies a deadlock that, in fact, occurs for a large class of controllers that x^* belongs to. We say a controller $x = (v_{\ell 0}, v_{r0}, v_{\ell 1}, v_{r1}) \in [-1, 1]^4$ is *clockwise-searching* if $v_{r0} < v_{\ell 0} < 0$. In other words, a clockwise-searching controller maps $I = 0$ (i.e., the case in which no robot is detected by the line-of-sight sensor) to a clockwise rotation about the center of rotation c that is a distance $R > 0$ away.[1]

Theorem 4. *For all $n > 3$ and all clockwise-searching controllers x, there exists an initial configuration of n robots from which the system will not aggregate when using controller x.*

Proof. At a high level, we construct a deadlocked configuration by placing the n robots in pairs such that no robot sees any other robot with its line-of-sight sensor—implying that all robots continually try to rotate about their centers of rotation—and each pair's robots mutually block each other's rotation. This suffices for the case that n is even; when n is odd, we extend the all-pairs configuration to include one mutually blocking triplet. Thus, no robots can move in this configuration since they are all mutually blocking, and since no robot sees any other they remain in this disconnected (non-aggregated) configuration indefinitely.

In detail, first suppose $n > 3$ is even. As in [21], let r denote the radius of a robot. For each $i \in \{0, 1, \dots, \frac{n}{2} - 1\}$, place robots p_{2i} and p_{2i+1} at points $(3r \cdot i, r)$ and $(3r \cdot i, -r)$, respectively. Orient all robots p_{2i} with their line-of-sight sensors in the $+y$ direction, and orient all robots p_{2i+1} in the $-y$ direction. This configuration is depicted in Fig. 1a. Due to their orientations, no robot can see any others; thus, since x is a clockwise-searching controller, all robots p_{2i} are attempting to move in the $-y$ direction while all robots p_{2i+1} are attempting to move in the $+y$ direction. Each pair of robots is mutually blocking, resulting in no motion. Moreover, since each consecutive pair of robots has a horizontal gap of distance r between them, this configuration is disconnected and thus non-aggregated.

It remains to consider when $n > 3$ is odd. Organize the first $n - 3$ robots in pairs according to the description above; since n is odd, we have that $n - 3$ must be even. Then place robot p_{n-1} at point $(3r(\frac{n}{2} - 1) + \sqrt{3}r, 0)$ with its line-of-sight sensor oriented at $0°$ (i.e., the $+x$ direction), robot p_{n-2} at point $(3r(\frac{n}{2} - 1), -r)$ with orientation $240°$, and robot p_{n-3} at point $(3r(\frac{n}{2} - 1), r)$ with orientation $120°$, as depicted in Fig. 1b. By a nearly identical argument to the one above, this configuration will also remain deadlocked and disconnected.

Therefore, in all cases there exists a configuration of $n > 3$ robots from which no clockwise-searching controller can achieve aggregation. □

[1] Note that an analogous version of Theorem 4 would hold for counter-clockwise-searching controllers if a robot's center of rotation was $90°$ clockwise rather than counter-clockwise from its line-of-sight sensor.

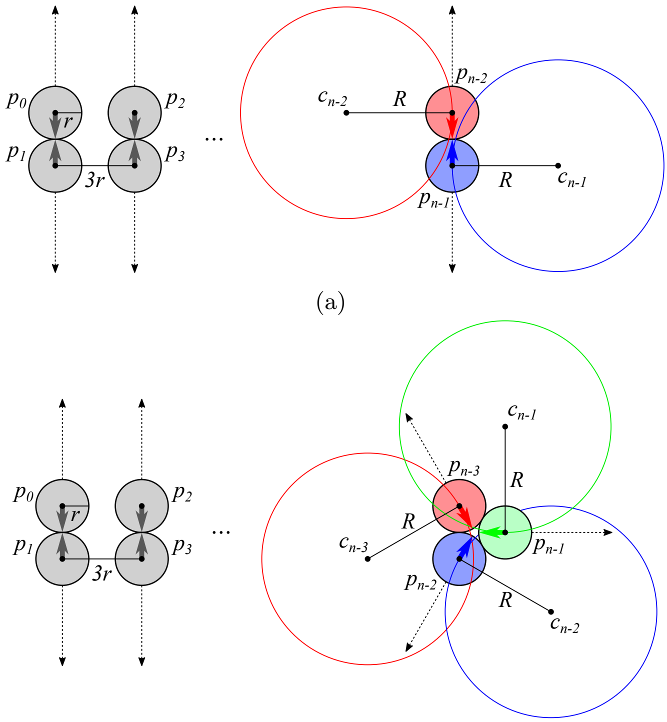


Fig. 1. The deadlocked configurations described in the proof of Theorem 4 for (a) $n > 3$ even and (b) $n > 3$ odd that remain non-aggregated indefinitely.

We have shown that no clockwise-searching controller (including x^*) can be guaranteed to aggregate a system of $n > 3$ robots starting from a deadlocked configuration, implying that Conjecture 1 does not hold in general. Moreover, not all deadlocked configurations are disconnected: Fig. 2 shows a connected configuration that will never make progress towards a more compact configuration because all robots are mutually blocked by their neighbors. Notably, these deadlocks are not observed in practice due to inherent noise in the physical e-puck robots. Real physics work to aggregation's advantage: if the robots were to ever get "stuck" in a deadlock configuration, collisions and slipping perturb the precise balancing of forces to allow the robots to push past one another. This motivates an explicit inclusion and modeling of noise in the algorithm, which we will return to in the next section.

Aaron Becker had conjectured at Dagstuhl Seminar 18331 [4] that symmetry could also lead to livelock, a second type of negative result for the Gauci et al. algorithm. In particular, Becker conjectured that robots initially organized in a cycle (e.g., Fig. 3a for $n = 3$) would traverse a "symmetric dance" in perpetuity without converging to an aggregated state when using controller x^*. However,

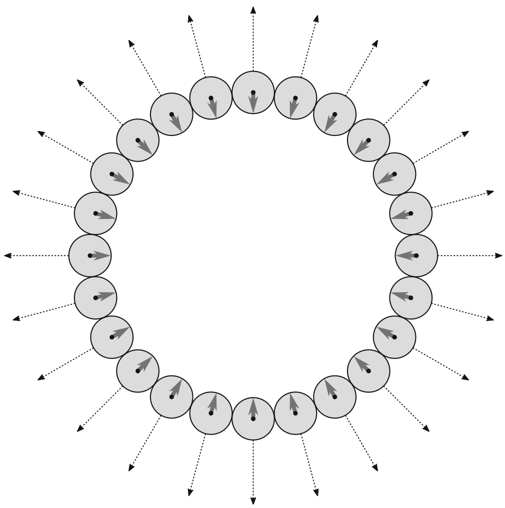


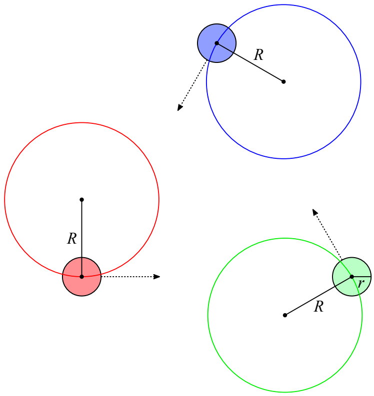
Fig. 2. A connected deadlocked configuration that remains non-compact.

simulations disprove this conjecture. Figure 3b shows that while swarms of various sizes initialized in the symmetric cycle configuration do exhibit an oscillatory behavior, they always reach and remain near the minimum dispersion value indicating near-optimal aggregation. Interestingly, these unique initial conditions cause small swarms to reach and remain in an oscillatory cycle where they touch and move apart infinitely often. Larger swarms break symmetry through collisions once the robots touch.

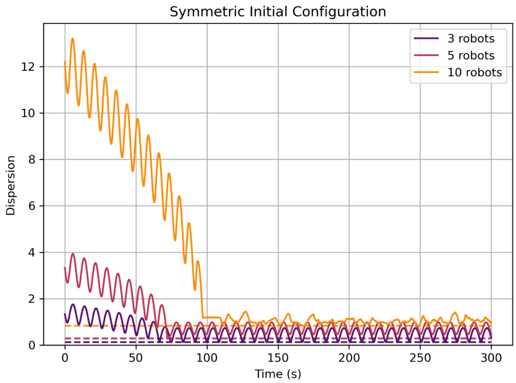
4 Robustness to Error and Noise

Motivated by the role of collisions and perturbations in freeing swarms from potential deadlocks, we next investigate the algorithm's *robustness* to varying magnitudes of error and noise. Our simulation platform models robots as circular rigid bodies in two dimensions, capturing all translation, rotation, and collision forces acting on the robots. Forces are combined and integrated iteratively over 5 ms time steps to obtain the translation and rotation of each robot. Figure 4 shows each of the four aggregation metrics for a baseline run on a swarm of $n = 100$ robots with no explicitly added noise. All four metrics demonstrate the system's steady but non-monotonic progress towards aggregation. Smallest enclosing disc circumference, convex hull perimeter, and dispersion show qualitatively similar progressions while the cluster fraction highlights when individual connected components join together.

We study the effects of two different forms of noise: *motion noise* and *error probability*. For motion noise, each robot at each time step experiences an applied force of a random magnitude in $[0, m^*]$ in a random direction. The parameter m^* defines the maximum noise force (in newtons) that can be applied to a robot



(a)



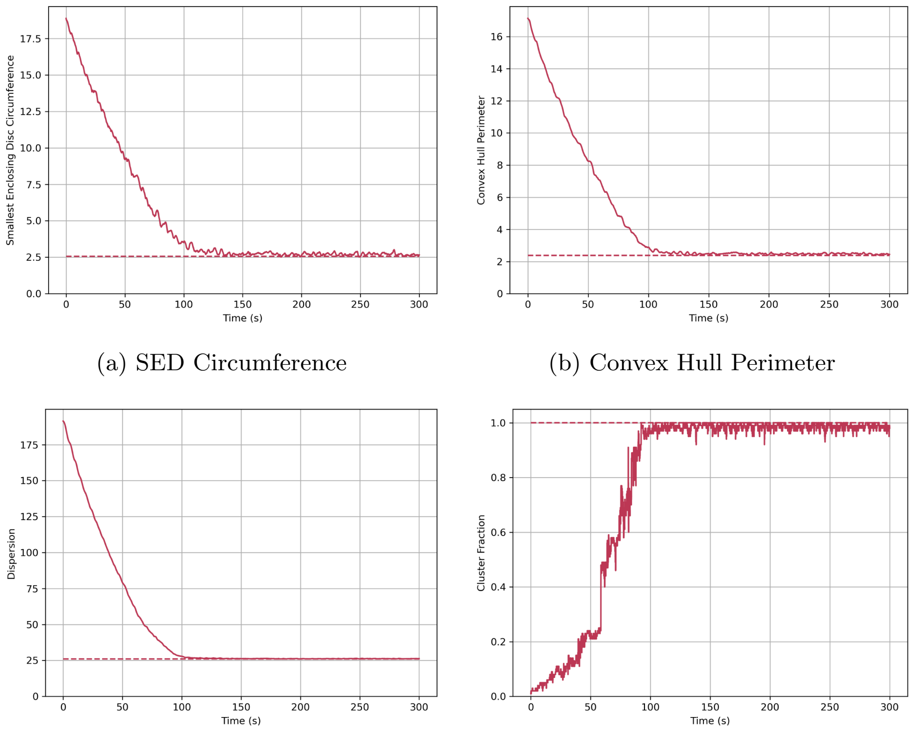
(b)

Fig. 3. (a) An example symmetric configuration of $n = 3$ robots that was conjectured to produce livelock. (b) Dispersion over time for swarms of $n = 3$ (purple), $n = 5$ (magenta), and $n = 10$ (orange) robots with symmetric initial configurations analogous to that of Fig. 3a. Dashed lines show the theoretical minimum dispersion value for the given system size. (Color figure online)

in a single time step. For error probability, each robot has the same probability $p \in [0, 1]$ of receiving the incorrect feedback from its sight sensor at each time step; more formally, a robot will receive the correct feedback I with probability $(1-p)$ and the opposite, incorrect feedback $1-I$ with probability p.[2] The robot then proceeds with the algorithm as usual based on the feedback it receives.

In general, as the magnitude of error increases, so does the time required to achieve aggregation. The algorithm exhibits robustness to low magnitudes of motion noise with the average time to aggregation remaining relatively steady for $m^* \leq 5$ N and increasing only minimally for $5 \leq m^* \leq 20$ N (Fig. 5a). With larger magnitudes of motion noise ($m^* > 20$ N), average time to aggregation increases significantly, with many runs reaching the limit for simulation time before aggregation is reached. A similar trend is evident for error probability (Fig. 5b). The algorithm exhibits robustness for small error probabilities $p \in [0, 0.05]$ with the average time to aggregation rising steadily with increased error until nearly all runs reach the simulation time limit. Intuitively, while small amounts of noise can help the algorithm overcome deadlock without degrading performance, too much noise interferes significantly with the algorithm's ability to progress towards aggregation.

[2] Our formulation of an "error probability" p is equivalent to "sensory noise" in [21] when the false positive and false negative probabilities are both equal to p.



(a) SED Circumference

(b) Convex Hull Perimeter

(c) Dispersion

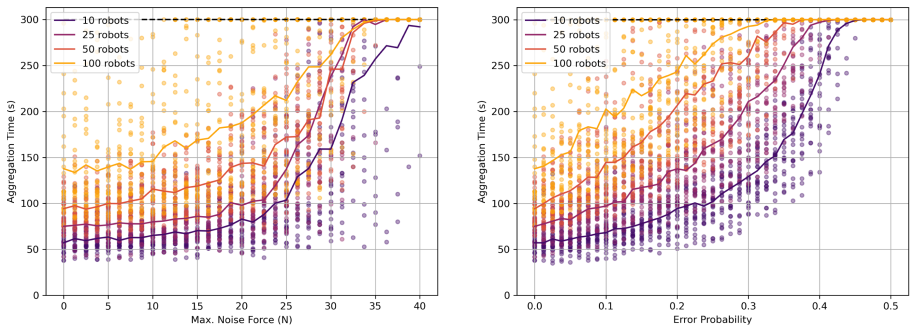
(d) Cluster Fraction

Fig. 4. Time evolutions of the four aggregation metrics for the same execution of x^* by a system of $n = 100$ robots for 300 s with no explicitly added noise. Dashed lines indicate the optimal value for each aggregation metric given the number of robots n.

5 Using a Cone-of-Sight Sensor

We next analyze a generalization of the algorithm where each robot has a *cone-of-sight sensor* of angle β instead of a line-of-sight sensor ($\beta = 0$). This was left as future work in [21] and was briefly considered in [20] where, for each $\beta \in \{0°, 30°, \ldots, 180°\}$, the best performing controller x_β was found via exhaustive search and compared against the others. Here we take a complementary approach, studying the performance of the original controller x^* as β varies.

We begin by proving that, in the case of one static robot and one robot executing the generalized algorithm, using a cone-of-sight sensor with size $\beta > 0$ can improve the time to aggregation by a linear factor (as a function of the initial distance between the two robots) over the original algorithm. This result follows from the fact that progress towards aggregation is achieved when the moving robot is rotating in place, moving its center of rotation closer to the static robot. With a line-of-sight sensor, the further the two robots are from each other, the smaller the moving robot's rotation in place. However, with a



(a) Motion Noise (b) Error Probability

Fig. 5. The time required to reach aggregation for different magnitudes of (a) motion noise and (b) error probability for systems of $n = 10$ (purple), $n = 25$ (magenta), $n = 50$ (red), and $n = 100$ (orange) robots. Each experiment for a given system size and noise strength was repeated 25 times (scatter plot); average runtimes are shown as solid lines. We consider systems that are within 15% of the minimum dispersion value as aggregated. The dashed line at 300 seconds indicates the cutoff time at which the run is determined to be non-aggregating. (Color figure online)

cone-of-sight sensor, the moving robot is guaranteed to rotate in place a fixed amount each time it sees the static robot, guaranteeing at least constant progress towards aggregation with each revolution.

Theorem 5. *One moving robot using a cone-of-sight sensor of size $\beta \in (0, \pi)$ will always aggregate with another static robot in*

$$m < \left\lceil \frac{(d_0 - R - r_i - r_j)(R + 2r_i)}{2\sqrt{3}Rr_i \sin((1 - 1/\sqrt{3}) \cdot \beta/2)} \right\rceil$$

rotations around its center of rotation, where d_0 is the initial value of $||\boldsymbol{p}_j - \boldsymbol{c}_i||$.

Proof. Consider a robot i executing the generalized algorithm at position $\boldsymbol{p}_i$ with center of rotation $\boldsymbol{c}_i$ and a static robot j at position $\boldsymbol{p}_j$. As in the proofs of Theorems 5.1 and 5.2 in [21], we first consider the scenario shown in Fig. 6 and derive an expression for $d' = ||\boldsymbol{p}_j - \boldsymbol{c}'_i||$ in terms of $d = ||\boldsymbol{p}_j - \boldsymbol{c}_i||$. W.l.o.g., let $\boldsymbol{c}_i = [0, 0]^T$ and let the axis of the cone-of-sight sensor of robot i point horizontally right at the moment it starts seeing robot j. Then the position of robot j is given by

$$\boldsymbol{p}_j = \begin{bmatrix} \frac{r_j \cos(\alpha/2+\gamma)}{\sin(\alpha/2)} \\ -\left(R + \frac{r_j \sin(\alpha/2+\gamma)}{\sin(\alpha/2)}\right) \end{bmatrix}.$$

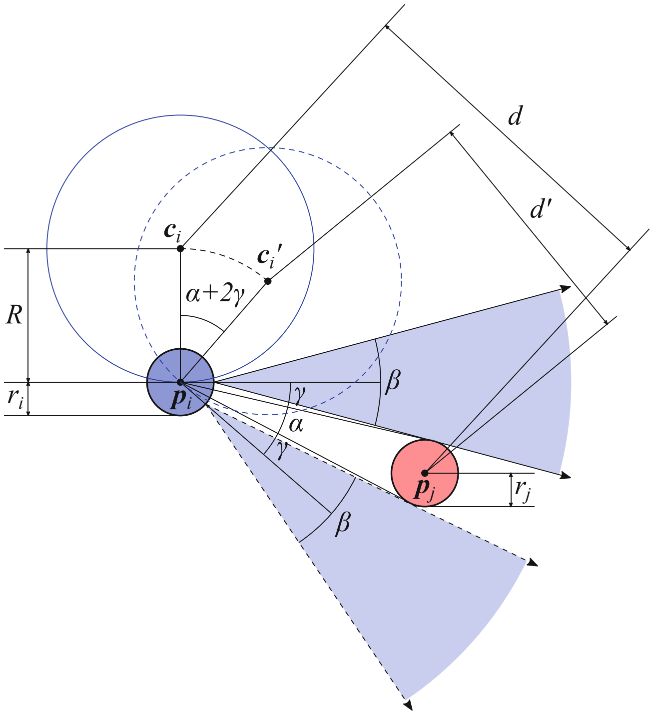


Fig. 6. The setup considered in the proof of Theorem 5. Robot i is moving and has a cone-of-sight sensor with size β while robot j is static.

Substituting this position into the distance $d = ||\boldsymbol{p}_j - \boldsymbol{c}_i||$ yields

$$d^2 = \left(\frac{r_j \cos(\alpha/2+\gamma)}{\sin(\alpha/2)}\right)^2 + \left(R + \frac{r_j \sin(\alpha/2+\gamma)}{\sin(\alpha/2)}\right)^2$$
$$= R^2 + \frac{2Rr_j \sin(\alpha/2+\gamma)}{\sin(\alpha/2)} + \frac{r_j^2}{\sin^2(\alpha/2)}.$$

Using a line-of-sight sensor, robot i would only rotate α before it no longer sees robot j; however, with a cone-of-sight sensor of size β, robot i rotates $\alpha + 2\gamma$ before robot j leaves its sight, where γ is the angle from the cone-of-sight axis to the first line intersecting $\boldsymbol{p}_i$ that is tangent to robot j. With this cone-of-sight sensor, $\boldsymbol{c}_i'$ is given by

$$\boldsymbol{c}_i' = \begin{bmatrix} R\sin(\alpha+2\gamma) \\ R(\cos(\alpha+2\gamma)-1) \end{bmatrix}.$$

Substituting this new center of rotation into the distance $d' = ||\boldsymbol{p}_j - \boldsymbol{c}'_i||$ yields

$$
\begin{aligned}
d' &= \sqrt{\begin{array}{l}\left(\dfrac{r_j \cos(\alpha/2+\gamma)}{\sin(\alpha/2)} - R\sin(\alpha+2\gamma)\right)^2 \\ + \left(-\left(R + \dfrac{r_j \sin(\alpha/2+\gamma)}{\sin(\alpha/2)}\right) - R(\cos(\alpha+2\gamma)-1)\right)^2\end{array}} \\
&= \sqrt{\begin{array}{l}\dfrac{r_j^2 \cos^2(\alpha/2+\gamma)}{\sin^2(\alpha/2)} - \dfrac{2Rr_j \cos(\alpha/2+\gamma)\sin(\alpha+2\gamma)}{\sin(\alpha/2)} + R^2\sin^2(\alpha+2\gamma) \\ + \dfrac{r_j^2 \sin^2(\alpha/2+\gamma)}{\sin^2(\alpha/2)} + \dfrac{2Rr_j \sin(\alpha/2+\gamma)\cos(\alpha+2\gamma)}{\sin(\alpha/2)} + R^2\cos^2(\alpha+2\gamma)\end{array}} \\
&= \sqrt{R^2 + \frac{r_j^2}{\sin^2(\alpha/2)} + \frac{2Rr_j(\sin(\alpha/2+\gamma)\cos(\alpha+2\gamma) - \cos(\alpha/2+\gamma)\sin(\alpha+2\gamma))}{\sin(\alpha/2)}} \\
&= \sqrt{d^2 + \frac{2Rr_j(\sin(\alpha/2+\gamma-(\alpha+2\gamma)) - \sin(\alpha/2+\gamma))}{\sin(\alpha/2)}} \\
&= \sqrt{d^2 - \frac{4Rr_j \sin(\alpha/2+\gamma)}{\sin(\alpha/2)}}.
\end{aligned}
$$

Note that this relation contains the result proven in Theorem 5.1 of [21] as a special case by setting $\beta = 0$ (and thus $\gamma = 0$), which corresponds to a line-of-sight sensor. To bound the number of $d \to d'$ updates required until $d \leq R + r_i + r_j$ (i.e., until the robots have aggregated), we write the following recurrence relation, where $\hat{d}_m = d_m^2$ and $\hat{d}_m > (R + r_i + r_j)^2$:

$$\hat{d}_{m+1} = \hat{d}_m - \frac{4Rr_j \sin(\alpha/2+\gamma)}{\sin(\alpha/2)}.$$

Observe that α is the largest when the two robots are touching, and—assuming $r_i = r_j$, i.e., the two robots are the same size—it is easy to see that $\alpha \leq \pi/3$. Also, γ is at least 0 and at most $\beta/2$; thus, by supposition, $\gamma < \pi/2$. Thus, by the angle sum identity,

$$\hat{d}_{m+1} < \hat{d}_m - \frac{4Rr_j \cos(\alpha/2)\sin(\gamma)}{\sin(\alpha/2)}.$$

Again, since $\alpha \leq \pi/3$, we have $\cos(\alpha/2) \geq \sqrt{3}/2$. By inspection, we also have $\sin(\alpha/2) < r_j/(d_m - R)$, yielding

$$\hat{d}_{m+1} < \hat{d}_m - 2\sqrt{3}R(d_m - R)\sin(\gamma).$$

Let $k_1 > 1$ be a constant such that $r_i = r_j = R/k_1$, which must exist since r_i, r_j, and R are constants and $r_i = r_j < R$. Since the robots have not yet aggregated, we have $d_m > R + r_i + r_j = (1 + 2/k_1)R$. We use this to show

$$R < \frac{d_m}{1 + 2/k_1} = k_2 d_m,$$

where $k_2 = 1/(1 + 2/k_1) < 1$ is a constant. Returning to our recurrence relation:

$$\hat{d}_{m+1} < \hat{d}_m - 2\sqrt{3}R(d_m - k_2 d_m)\sin(\gamma) = \hat{d}_m - 2\sqrt{3}R d_m(1 - k_2)\sin(\gamma)$$

Recalling that $\hat{d}_m = d_m^2$, we have

$$d_{m+1} < \sqrt{\hat{d}_m - 2\sqrt{3}R d_m(1 - k_2)\sin(\gamma)} < d_m - \sqrt{3}R(1 - k_2)\sin(\gamma),$$

where the second inequality can be verified by squaring both sides and noting that $R > 0$, $1 - k_2 > 0$, and $\gamma \in (0, \pi/2)$. As a final upper bound on d_{m+1}, we lower bound the angle γ as a function of the constant size of the cone-of-sight sensor β as $\gamma \geq (1 - 1/\sqrt{3}) \cdot \beta/2$ (see [10] for a complete derivation), yielding

$$d_{m+1} < d_m - \sqrt{3}R(1 - k_2)\sin((1 - 1/\sqrt{3}) \cdot \beta/2).$$

This yields the solution

$$d_m < d_0 - m\sqrt{3}R(1 - k_2)\sin((1 - 1/\sqrt{3}) \cdot \beta/2), \quad d_m > R + r_i + r_j$$

The number of $d \to d'$ updates required until $d \leq R + r_i + r_j$ is now given by setting $d_m = R + r_i + r_j$ in this solution and solving for m, which yields

$$m < \left\lceil \frac{d_0 - R - r_i - r_j}{\sqrt{3}R(1 - k_2)\sin((1 - 1/\sqrt{3}) \cdot \beta/2)} \right\rceil = \left\lceil \frac{(d_0 - R - r_i - r_j)(R + 2r_i)}{2\sqrt{3}R r_i \sin((1 - 1/\sqrt{3}) \cdot \beta/2)} \right\rceil,$$

concluding the proof. □

We note that this bound on the number of required updates m has a linear dependence on d_0 while the original bound proven in Theorem 5.2 of [21] for line-of-sight sensors depended on d_0^2, demonstrating a linear speedup with cone-of-sight sensors for $n = 2$ robots. However, simulation results show that as the number of robots increases, the speedup from using cone-of-sight sensors diminishes (Fig. 7). All systems benefit from small cone-of-sight sensors—i.e., $\beta \in (0, 0.5)$—reaching aggregation in significantly less time. With larger systems, however, large cone-of-sight sensors can be detrimental as robots see others more often than not, causing them to primarily rotate in place without making progress towards aggregation. This highlights a delicate balance between the algorithm's two modes (rotating around the center of rotation and rotating in place) with β indirectly affecting how much time is spent in each.

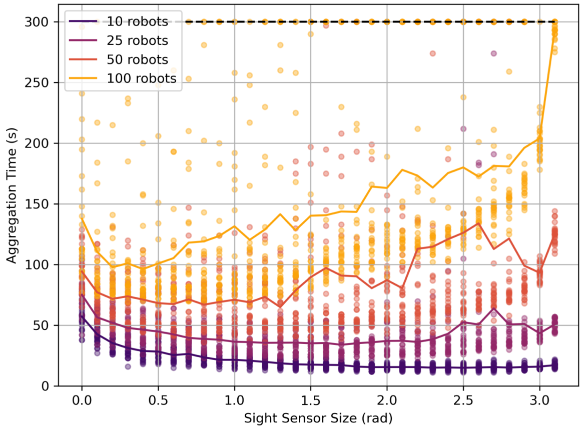


Fig. 7. The effects of cone-of-sight sensor size on the algorithm's time to aggregation for systems of $n = 10$ (purple), $n = 25$ (magenta), $n = 50$ (red), and $n = 100$ (orange) robots. Each experiment for a given system size and sensor size was repeated 25 times (scatter plot); average runtimes are shown as solid lines. We consider systems that are within 15% of the minimum dispersion value as aggregated. The dashed line at 300 seconds indicates the cutoff time at which the run is determined to be non-aggregating. (Color figure online)

6 Conclusion

In this paper, we investigated the Gauci et al. swarm aggregation algorithm [21] which provably aggregates two robots and reliably aggregates larger swarms in experiment using only a binary line-of-sight sensor and no arithmetic computation or persistent memory. We answered the open question of whether the algorithm guarantees aggregation for systems of $n > 2$ robots negatively, identifying how deadlock can halt the system's progress towards aggregation. In practice, however, the physics of collisions and slipping work to the algorithm's advantage in avoiding deadlock; moreover, we showed that the algorithm is robust to small amounts of noise in its sensors and in its motion. Finally, we considered a generalization of the algorithm using cone-of-sight sensors, proving that for the situation of one moving robot and one static robot, the time to aggregation is improved by a linear factor over the original line-of-sight sensor. Simulation results showed that small cone-of-sight sensors can also improve runtime for larger systems, though with diminishing returns.

In the full version of this work [10], we additionally introduced a noisy, discrete adaptation of the Gauci et al. algorithm in an effort to formally prove its convergence when noise is explicitly modeled as a mechanism to break deadlock. However, both the original algorithm and this discrete adaptation progress towards aggregation non-monotonically, complicating analysis techniques relying on consistent progress towards the goal state. It is possible that a proof showing convergence *in expectation* can be derived, but we leave this for future work.

Acknowledgements and Data Availability. We thank Dagstuhl [4] for hosting the seminar that inspired this research, Roderich Groß for introducing us to this open problem, and Aaron Becker and Dan Halperin for their contributions to the investigations of symmetric livelock and cone-of-sight sensors. Source code for all simulations reported in this work is openly available at https://github.com/SOPSLab/SwarmAggregation.

References

1. Agrawal, M., Bruss, I.R., Glotzer, S.C.: Tunable emergent structures and traveling waves in mixtures of passive and contact-triggered-active particles. Soft Matter **13**(37), 6332–6339 (2017)
2. Angluin, D., Aspnes, J., Diamadi, Z., Fischer, M.J., Peralta, R.: Computation in networks of passively mobile finite-state sensors. Distrib. Comput. **18**(4), 235–253 (2006)
3. Bayindir, L.: A review of swarm robotics tasks. Neurocomputing **172**, 292–321 (2016)
4. Berman, S., Fekete, S.P., Patitz, M.J., Scheideler, C.: Algorithmic foundations of programmable matter (Dagstuhl Seminar 18331). Dagstuhl Rep. **8**(8), 48–66 (2019)
5. Brambilla, M., Ferrante, E., Birattari, M., Dorigo, M.: Swarm robotics: A review from the swarm engineering perspective. Swarm Intell. **7**(1), 1–41 (2013)
6. Camazine, S., Franks, N.R., Sneyd, J., Bonabeau, E., Deneubourg, J.L., Theraula, G.: Self-Organization in Biological Systems. Princeton University Press, Princeton, NJ, USA (2001)
7. Chen, J., Gauci, M., Price, M.J., Groß, R.: Segregation in swarms of e-puck robots based on the Brazil nut effect. In: Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems, vol. 1, pp. 163–170 (2012)
8. Cieliebak, M., Flocchini, P., Prencipe, G., Santoro, N.: Solving the robots gathering problem. In: Baeten, J.C.M., Lenstra, J.K., Parrow, J., Woeginger, G.J. (eds.) ICALP 2003. LNCS, vol. 2719, pp. 1181–1196. Springer, Heidelberg (2003). https://doi.org/10.1007/3-540-45061-0_90
9. Cortés, J., Martinez, S., Bullo, F.: Robust rendezvous for mobile autonomous agents via proximity graphs in arbitrary dimensions. IEEE Trans. Autom. Control **51**(8), 1289–1298 (2006)
10. Daymude, J.J., Harasha, N.C., Richa, A.W., Yiu, R.: Deadlock and noise in self-organizing aggregation without computation (2021). https://arxiv.org/abs/2108.09403
11. Deblais, A., et al.: Boundaries control collective dynamics of inertial self-propelled robots. Phys. Rev. Lett. **120**(18), 188002 (2018)
12. Deneubourg, J.L., Grégoire, J.C., Le Fort, E.: Kinetics of larval gregarious behavior in the bark beetle Dendroctonus micans (Coleoptera: Scolytidae). J. Insect Behav. **3**, 169–182 (1990)
13. Devreotes, P.: Dictyostelium discoideum: A model system for cell-cell interactions in development. Science **245**(4922), 1054–1058 (1989)
14. Dorigo, M., Theraulaz, G., Trianni, V.: Reflections on the future of swarm robotics. Sci. Robot. **5**(49), eabe4385 (2020)
15. Dorigo, M., Theraulaz, G., Trianni, V.: Swarm robotics: Past, present, and future. Proc. IEEE **109**(7), 1152–1165 (2021)
16. Fatès, N.: Solving the decentralised gathering problem with a reaction-diffusion-chemotaxis scheme. Swarm Intell. **4**(2), 91–115 (2010)

17. Firat, Z., Ferrante, E., Gillet, Y., Tuci, E.: On self-organised aggregation dynamics in swarms of robots with informed robots. Neural Comput. Appl. **32**(17), 13825–13841 (2020). https://doi.org/10.1007/s00521-020-04791-0
18. Flocchini, P., Prencipe, G., Santoro, N. (eds.): Distributed Computing by Mobile Entities. Springer International Publishing, Switzerland (2019)
19. Flocchini, P., Prencipe, G., Santoro, N., Widmayer, P.: Gathering of asynchronous robots with limited visibility. Theor. Comput. Sci. **337**(1–3), 147–168 (2005)
20. Gauci, M.: Swarm Robotic Systems with Minimal Information Processing. PhD Thesis, University of Sheffield, Sheffield, England (2014). https://etheses.whiterose.ac.uk/7569/
21. Gauci, M., Chen, J., Li, W., Dodd, T.J., Groß, R.: Self-organized aggregation without computation. Int. J. Robot. Res. **33**(8), 1145–1161 (2014)
22. Graham, R.L., Sloane, N.J.A.: Penny-packing and two-dimensional codes. Discret. Comput. Geom. **5**(1), 1–11 (1990). https://doi.org/10.1007/BF02187775
23. Groß, R., Magnenat, S., Mondada, F.: Segregation in swarms of mobile robots based on the Brazil nut effect. In: 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 4349–4356. IROS 2009 (2009)
24. Hamann, H.: Swarm Robotics: A Formal Approach. Springer, Heidelberg (2018)
25. Jeanson, R., et al.: Self-organized aggregation in cockroaches. Anim. Behav. **69**(1), 169–180 (2005)
26. Li, S., et al.: Programming active cohesive granular matter with mechanically induced phase changes. Sci. Adv. **7**(17), eabe8494 (2021)
27. Magurran, A.E.: The adaptive significance of schooling as an anti-predator defence in fish. Ann. Zool. Fenn. **27**(2), 51–66 (1990)
28. Misir, O., Gökrem, L.: Dynamic interactive self organizing aggregation method in swarm robots. Biosystems **207**, 104451 (2021)
29. Mlot, N.J., Tovey, C.A., Hu, D.L.: Fire ants self-assemble into waterproof rafts to survive floods. Proc. Natl. Acad. Sci. **108**(19), 7669–7673 (2011)
30. Mondada, F., et al.: The e-puck, a robot designed for education in engineering. In: Proceedings of the 9th Conference on Autonomous Robot Systems and Competitions, pp. 59–65 (2009)
31. Özdemir, A., Gauci, M., Kolling, A., Hall, M.D., Groß, R.: Spatial coverage without computation. In: International Conference on Robotics and Automation, pp. 9674–9680 (2019)
32. Özedmir, A., Gauci, M., Bonnet, S., Groß, R.: Finding consensus without computation. IEEE Robot. Autom. Lett. **3**(3), 1346–1353 (2018)
33. Woods, D., Chen, H.L., Goodfriend, S., Dabby, N., Winfree, E., Yin, P.: Active self-assembly of algorithmic shapes and patterns in polylogarithmic time. In: Proceedings of the 4th Innovations in Theoretical Computer Science Conference, pp. 353–354 (2013)
34. Yu, J., LaValle, S.M., Liberzon, D.: Rendezvous without coordinates. IEEE Trans. Autom. Control **57**(2), 421–434 (2012)
35. Zebrowski, P., Litus, Y., Vaughan, R.T.: Energy efficient robot rendezvous. In: Fourth Canadian Conference on Computer and Robot Vision, pp. 139–148 (2007)