Gathering Anonymous, Oblivious Robots on a Grid

Matthias Fischer Daniel Jung Friedhelm Meyer auf der Heide

Heinz Nixdorf Institute & Computer Science Department
University of Paderborn (Germany)
Fürstenallee 11, 33102 Paderborn

{mafi,daniel.jung,fmadh}@uni-paderborn.de

Abstract

We consider a swarm of n autonomous mobile robots, distributed on a 2-dimensional grid. A basic task for such a swarm is the gathering process: all robots have to gather at one (not predefined) place. The work in this paper is motivated by the following insight: On one side, for swarms of robots distributed in the 2-dimensional Euclidean space, several gathering algorithms are known for extremely simple robots that are oblivious, have bounded viewing radius, no compass, and no "flags" to communicate a status to others. On the other side, in case of the 2-dimensional grid, the only known gathering algorithms for robots with bounded viewing radius without compass, need to memorize a constant number of rounds and need flags.

In this paper we contribute the, to the best of our knowledge, first gathering algorithm on the grid, which works for anonymous, oblivious robots with bounded viewing range, without any further means of communication and without any memory. We prove its correctness and an $O(n^2)$ time bound. This time bound matches those of the best known algorithms for the Euclidean plane mentioned above.

Keywords: gathering problem, autonomous robots, distributed algorithms, local algorithms, mobile agents, runtime bound, swarm formation problems

1 Introduction

Swarm robotics considers large swarms of relatively simple mobile robots deployed to some two- or three-dimensional area. These robots have very limited sensor capabilities; typically they can only observe aspects of their local environment. The objective of swarm robotics is to understand which global behavior of a swarm is induced by local strategies, simultaneously executed by the individual robots. Typically, the decisions of the individual robots are based on local information only.

In order to formally argue about the impact of such local decisions of the robots on the overall behavior of the swarm, many simple models of robots, their local algorithms, the space they live in, and underlying time models are proposed; for a survey see the book [FPS12] by Flocchini, Prencipe, and Santoro.

A basic desired global behavior of such a swarm is the gathering process: all robots have to gather at one (not predefined) place. Local algorithms for this process are defined and analysed for a variety of models. The work in this paper is motivated by the following insight: Consider the "standard" models for the environment, namely the Euclidean plane and its discretization as a 2-dimensional grid, and robots restricted to a local view: they can only "see" other robots within a

distance d, a fixed value defining the viewing radius. (In the plane, distance means the Euclidean distance, in the grid the Manhattan distance.) Further assume the fully synchronous time model $\mathcal{FSYNC}$ (see [ASY95, FPS12]). The current state of the art (see Section 2 for more details) shows the following: In the plane there are several algorithms for gathering known. They need $\mathcal{O}(n^2)$ synchronous steps for gathering [DKL+11]. The robot capabilities needed here are very rudimentary: a robot has to be able to compute the positions of the robots within its viewing range, relative to its own one. So no global coordinate system, no compass is needed. Further, the robots are anonymous and fully oblivious, i.e. they decide fully independent on past decisions. Also the need no communication besides viewing positions of their neighbors, especially they do not need flags or lights to communicate a status to others. Interestingly, in case of the 2-dimensional grid, there are no gathering strategies known at all for the case of robots being as simple as the one sufficient for gathering in the plane. In the only known algorithms on the grid for anonymous robots, the robots need flags to communicate states to neighbors, and have to be able to memorize a fixed number of steps. Interestingly, such capabilities even allow for gathering strategies running in $\mathcal{O}(n)$ rounds [CFJM16, ACF+16].

In this paper we contribute the, to the best of our knowledge, first such gathering algorithm on the grid, which works for anonymous, oblivious robots without any communication with flags. We prove its correctness and an $\mathcal{O}(n^2)$ time bound. More precisely, the running time depends quadratically on the outer boundary length of the swarm. The outer boundary is the seamless sequence of neighbouring robots that encloses all the others robots inside.

2 Related work

There is vast literature on robot problems, researching how specific coordination problems can be solved by a swarm of robots given a certain limited set of abilities. The robots are usually point-shaped (hence collisions are neglected) and positioned in the Euclidean plane. They can be equipped with a memory or are *oblivious*, i.e., the robots do not remember anything from the past and perform their actions only on their current views. If robots are anonymous, they do not carry any IDs and cannot be distinguished by their neighbors. Another type of constraint is the compass model: If all robots have the same coordinate system, some tasks are easier to solve than if all robots' coordinate systems are distorted. In [KTI+07, ISK+12] a classification of these two and also of dynamically changing compass models, as well as their effects regarding the gathering problem in the Euclidean plane, is considered. The operation of a robot is considered in the *look-compute-move model* [CP04]. How the steps of several robots are aligned is given by the *time model*, which can range from an asynchronous $\mathcal{ASYNC}$ model (for example, see [CP04]), where even the single steps of the robots' steps may be interleaved, to a fully synchronous $\mathcal{FSYNC}$ model (for example, see [ASY95]), where all steps are performed simultaneously. A collection of recent algorithmic results concerning distributed solving of basic problems like gathering and pattern formation, using robots with very limited capabilities, can be found in [FPS12].

One of the most natural problems is to gather a swarm of robots in a single point. Usually, the swarm consists of point-shaped, oblivious, and anonymous robots. The problem is widely studied in the Euclidean plane. Having point-shaped robots, collisions are understood as merges/fusions of robots and interpreted as gathering progress [DKM10]. In [CFPS03] the first gathering algorithm for the $\mathcal{ASYNC}$ time model with multiplicity detection (i.e., a robot can detect if other robots are also located at its own position) and global views is provided. Gathering in the local setting was studied in [ASY95]. In [Pre07] situations when no gathering is possible are studied. The question of gathering on graphs instead of gathering in the plane was considered in [Mar09, DFKP06,

KMP08]. In [SN13] the authors assume global vision, the $\mathcal{ASYNC}$ time model and furthermore allow unbounded (finite) movements. They show optimal bounds concerning the number of robot movements for special graph topologies like trees and rings.

Concerning the gathering on grids, in [DDSKN12] it is shown that multiplicity detection is not needed and the authors further provide a characterization of solvable gathering configurations on finite grids. In [SN14], these results are extended to infinite grids, assuming global vision. The authors characterize *gatherable* grid configurations concerning exact gathering in a single point. Under their robot model and the $\mathcal{ASYNC}$ time model, the authors present an algorithm which gathers *gatherable* configurations optimally concerning the total number of movements.

Assuming only local capabilities of the robots, esp. only local vision and no compass, makes gathering challenging. For example, with a given global vision or alternatively just the knowledge of a global compass, the robots could compute the center of the globally smallest enclosing square or circle and just move to this point (global vision) or all robots with local neighbors in front of them could simply move to the south-eastern direction, for example, and would finally meet (global compass).

In the $\mathcal{FSYNC}$ time model, the total running time is a quality measurement of an algorithm. In this time model exist several results that prove runtime bounds. For local robot models, this strongly restricts the robot capabilities: no global control, no unique ids, no compass, only local vision (i.e., they can only see other robots up to a constant distance) and no global communication.

But even under this strongly restricted model, the presence of remaining capabilities like allowing a constant number of states, constant memory or local communication, can drastically change running times by even more than the factor n. The price for this improvement then are much more complicated strategies. We face existing strategies concerning these parameters: We consider two basic formation problems: Maintaining and shortening a communication chain between an explorer and a base camp, and call them *chain strategies*. Starting with strategies in the Euclidean plane, there is the complex *Hopper* chain strategy [KM09] that works in time $\mathcal{O}(n)$, but needs robots with a constant number of states, a constant memory and local communication. Without these additional robot capabilities, the simple *Go-To-The-Middle* strategy [DKLMadH06] needs notably more time $\mathcal{O}(n^2 \log(n))$ for solving the same problem. Concerning the gathering under this restricted model, the simple *Got-To-The-Center* strategy [DKL+11] needs time $\mathcal{O}(n^2)$. (A faster strategy for the Euclidean plane does not exist, yet, and it is still unknown if this bound is tight.)

On the grid, the following strategies exist: Concerning chain strategies, the complex *Manhatten Hopper* [KM09] requires robots with a constant number of states, a constant memory and local communication, that works in time $\mathcal{O}(n)$. For the gathering, we have two quite complex strategies. One gathers a closed chain of robots [ACF+16], the other one an arbitrary connected swarm [CFJM16]. Both have asymptotically optimal running times of $\mathcal{O}(n)$ and the robots need a constant number of states, constant memory and local communication.

In the present paper, we instead only allow stateless, oblivious robots without communication. For this model, we deliver a simple strategy that gathers in time $\mathcal{O}(|\text{outer boundary}|^2) \subseteq \mathcal{O}(n^2)$, where $|\text{outer boundary}|$ denotes the length of the swarm's outer boundary (cf. Figure 5) which naturally is $\in \mathcal{O}(n)$. This is comparable to the $\mathcal{O}(n^2)$ bound for the Euclidean gathering [DKL+11] as there, the robots are also oblivious, stateless and cannot communicate. We suppose that $\mathcal{O}(|\text{outer boundary}|^2)$ is also tight in this model.

3 Our Local Grid Model

Our mobile robots need very few and simple capabilities: A robot moves on a two-dimensional grid and can change its position to one of its eight horizontal, vertical or diagonal neighboring grid points. It can see other robots only within a constant *viewing radius* of 7 (measured in L_1-distance). We call the range of visible robots *viewing range*. Within this viewing range, a robot can only see the relative positions of the included robots. The robots have no compass, no global control, no IDs. They cannot communicate. They do not have any states and are oblivious.

Our algorithm uses the fully synchronous time model $\mathcal{FSYNC}$, in that all robots are always active and do everything synchronously. Time is subdivided into equally sized rounds of constant lengths. In every round all robots simultaneously execute their operations in the common *look-compute-move model* [CP04], which divides one operation into three steps. Every round contains only on cycle of these steps: In the *look step*, the robot gets a snapshot of the current scenario from its own perspective, restricted to its constant-sized viewing range. During the *compute step*, the robot computes its action, and eventually performs it in the *move step*.

We say gathering is done, if all robots are located within a 2×2 square, because such configurations cannot be solved in our time model.

The swarm must be connected. In out model, two robots are connected, if they are located in horizontal on vertical neighboring grid cells. The operations of our algorithm do not destroy connectivity.

4 The algorithm

A robot can just depending on the current robot positions within its viewing-range decide to hop on one of its 8 neighboring grid cells. We distinguish diagonal (Subsection 4.1) and horizontal/vertical (Subsection 4.2) hops. The hops are intended to achieve gathering progress by modifying then swarm's *outer boundary*. Figure 5 defines the swarm's boundaries: Black and hatched robots are *boundary* robots. Black robots are further located on the swarm's *outer boundary*. In the figure, all other robots are grey colored. White cells are empty.

4.1 Diagonal hops

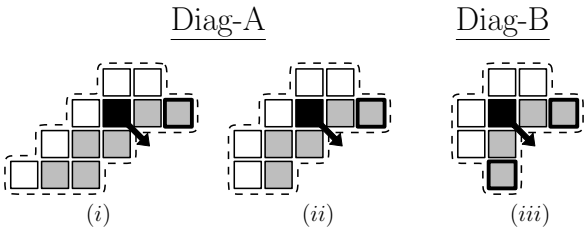

Figure 1: **Hop patterns:** One of the Diag-{A, B} hop patterns must match with the relative robot positions within the black robot's viewing-range. This is the Hop criterion for allowing the black robot perform the depicted diagonal hop.

If a robot checks if it can execute a diagonal hop, it compares the patterns of Figure 1 and 2 to the robot positions in its viewing range. It hops only, if it finds a matching Hop pattern but no matching, arbitrarily rotated and mirrored Inhibit patterns. For this, it first searches a matching

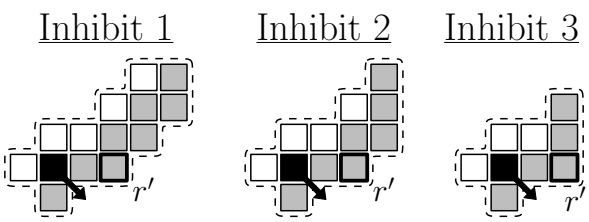

Figure 2: **Inhibit patterns:** Patterns that, in case they match, inhibit the black robot's hop.

Hop pattern (at most one can fit). And if it has found it, then it checks, if it can also find a matching Inhibit pattern under the constraint, that its own position (the black marked robot) is fixed and furthermore the position of the fat framed robots must be congruent. In case, that the matching Hop pattern is Diag-B, the hop is only inhibited, if for both fat framed robots some Inhibit pattern was found.

If a robot has hopped onto an occupied grid cell, the robots from then behave like one robot. We say, they *merge* and remove one of them.

4.2 Horizontal/vertical (HV) hops

Robots can also hop in vertical or horizontal direction. Then, we allow these hops for length 1 and 2 (cf. Figure 3). For length 2, horizontal respectively vertical hops are a joint operation of two robots. If for a robot a horizontal and a vertical HV hop apply at the same time ($b^\star$)), then it instead performs a diagonal hop as shown in the figure. After a HV hop, every target cell contains

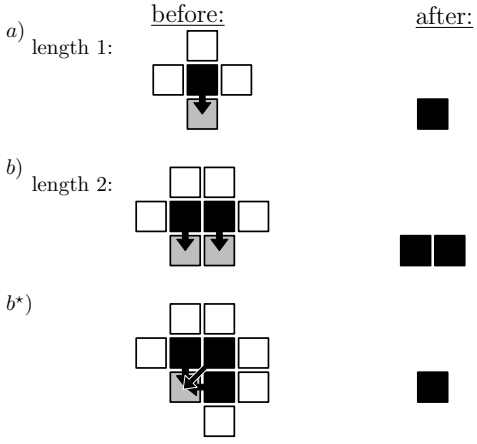

Figure 3: The black robots simultaneously hop downwards. Afterwards, duplicate robots *merge*.

at least two robots. We let these duplicate robots *merge*, i.e., we remove all but one of the robots of the according cell.

Diag-{A, B} and HV hops are executed simultaneously in the same step of the algorithm. Summarizing, all robots synchronously execute the algorithm, shown in Figure 4.

Every robot r in every round does the following:

Look: r takes a snapshot of the relative robot positions within its viewing range.
Compute: r checks which of the HV hop, Diag-$\{A, B\}$ hop and Inhibition patterns match the snapshot. Based on the matching patterns, r decides if it either does not hop or which neigboring grid cell it will hop.
Move: If r has calculated a hop, it now is executed. Else, r does not move.

Figure 4: The algorithm.

5 Measuring the gathering progress

The progress of our strategy is hardly based on the length of the swarm's outer boundary. We now, give more detailed definitions:

In Figure 5 that are black or hatched, are boundary robots. The black ones furthermore belong to the swarm's outer boundary, which is the boundary that borders on the outside of the swarm. In the figure, all other robots are grey colored.

We measure the outer boundary's length as follows: We start at a cell of the outer boundary and perform a complete walk along this boundary while we define the *length* as the total number of steps that we performed during this walk. That means, that, e.g. if the swarm is hourglass shaped, some robots are counted multiple (up to four) times. We denote this length by "|outer boundary|".

When speaking about a *subboundary*, we mean a connected sequence of robots of some boundary.

On the boundary, we further distinguish *convex* and *concave vertices*. In Figure 6 fat curves mark *convex vertices* of the swarm's outer boundary, while the thin curves mark the *concave vertices* of the swarm's outer boundary.

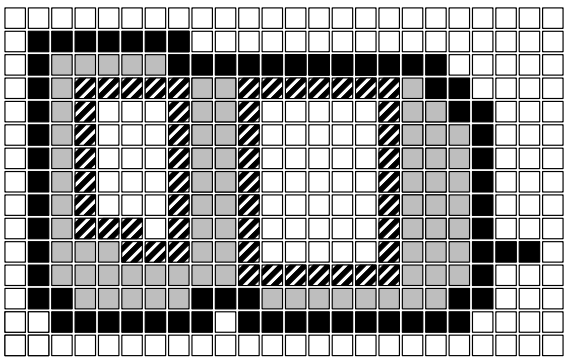

Figure 5: Definition of the (*outer*) *boundary*; Outer boundary: black robots

Outline of the running time proof For estimating the total running time, we need three different progress measurements: The first two of them are monotonically decreasing: The length of the swarm's outer boundary and the difference between the maximum possible number of convex vertices on the outer boundary and its actual value. We estimate the number of rounds in which both of them do not decrease, by the size of the area that is included in the swarm's outer boundary. Though this measurement is not monotone decreasing, we prove that at least in every round without boundary and convex progress, the area decreases by some constant size. We upper bound the size

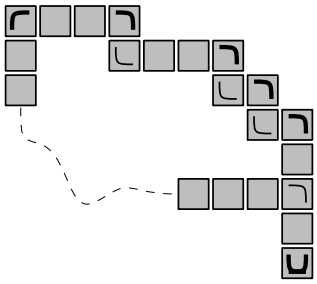

Figure 6: Definition of *convex* and *concave vertex*. Convex vertices: fat curves

by that the area can instead be increased during other rounds and show that this makes the total running time worse at most by a constant factor. While boundary and convex can have progress only $\mathcal{O}(|\text{outer boundary}|)$ times, area needs up to $\mathcal{O}(|\text{outer boundary}|^2)$ times. This then leads to the total running time $\mathcal{O}(|\text{outer boundary}|^2)$.

Progress measurement in more detail We distinguish three kinds of progress measures that help us prove the quadratic running time.

***Boundary*:** Length of the swarm's outer boundary.
***Convex*:** Difference between the number of convex vertices on the swarm's outer boundary and its maximum value.
***Area*:** Included area.

We have designed the hops this way, that the length of the outer boundary never increases. But it can remain unchanged over several rounds. Then, instead, we measure the progress by *Convex*. As we draw robots as squares, the total number of convex vertices on the swarms outer boundary is naturally upper bounded by 4|outer boundary|. *Convex* is the difference between this upper bound and the actual number of convex vertices on the outer boundary. We will show that also *Convex* never increases.

In rounds in that both, *Boundary* and *Convex*, do not achieve progress, we measure the gathering progress by *Area*. We measure *Area* as the number of robots on the swarm's outer boundary plus the number of inside cells (occupied as well as empty ones). In contrast to the other progress measurements, *Area* does not behave monotone in general, but is monotone in rounds without *Boundary* and *Convex* progress.

All three measurements only depend on the length of the swarm's outer boundary. While *Boundary* and *Convex* are linear, *Area* is quadratic. This then leads us to a total running time $\mathcal{O}(|\text{outer boundary}|^2)$.

6 Correctness & running time

In this section, we formally prove the correctness of the progress measurements and finally the total running time (Theorem 1).

6.1 Progress measurement *Boundary*

Lemma 1. *During the whole gathering,* Boundary *progress is monotonically increasing.*

Proof. As the definition of HV hops requires that the robots hop onto occupied cells, such hops naturally cannot increase the number of robots on the outer boundary. So we consider Diag-{A, B} hops in which robots hop towards the swarm's outside. In order to increase the number of robots on the outer boundary, the target cell of such hops must be empty. But then, the hopped robot has also been part of the outer boundary before the hop. □

6.2 Progress measurement *Convex*

Lemma 2. *During the whole gathering,* Convex *progress is monotonically increasing.*

Proof. If we say, "convex/concave vertices", we only consider the outer boundary. First, we analyze the HV hops. Here, a HV hop can reduce the number of convex vertices by at most 2. At the same time, the outer boundary becomes shorter by at least 2 robots. Then, our progress definition ensures that *Convex* either remains unchanged or increases.

Concerning the Diag-{A, B} hops, we look at Figure 7: The figure shows how the diagonal hops

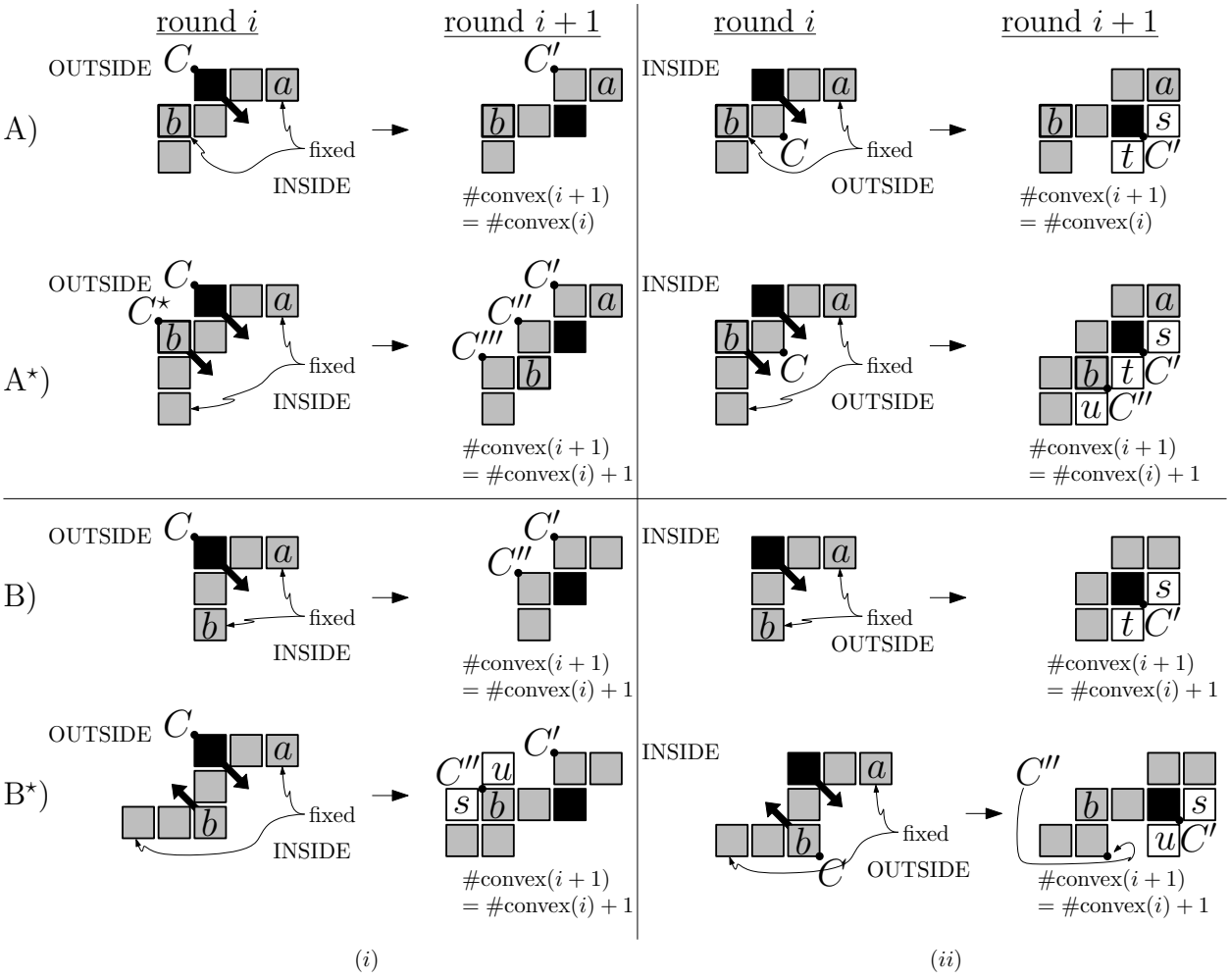

Figure 7: Local effect of all kinds of hops on the number of convex vertices on the outer boundary.

can (locally) change the number of convex vertices on the outer boundary. (In the figure, (ii) shows the hops from (i), but for switched INSIDE and OUTSIDE.) In all cases, the Inhibition patterns

ensure that the robot a does not move. In the figure, we distinguish Diag-A and Diag-B hops, while $\{\mathrm{A}, \mathrm{A}^\star\}$ refer to Diag-A and $\{\mathrm{B}, \mathrm{B}^\star\}$ to Diag-B. We distinguish the case that the robot b does not hop ($\{\mathrm{B}, \mathrm{B}\}$) and the other case, that it performs a hop ($\{\mathrm{A}^\star, \mathrm{B}^\star\}$).

The result of the case distinction is that in column (i) the number of convex vertices never decreases. In column (ii) this is also the case if the white marked cells (s, t, u) are empty.

In (ii), this is not the case, if some of the white marked cells (s, t, u) contain robots. We now show, that still the *Convex* progress isn't reversed, because instead the outer boundary becomes shorter. Then, also the maximum value for the total number of convex vertices becomes smaller, so that in our measurement the *Convex* progress is not reversed: Figure 8 shows the relevant cases.

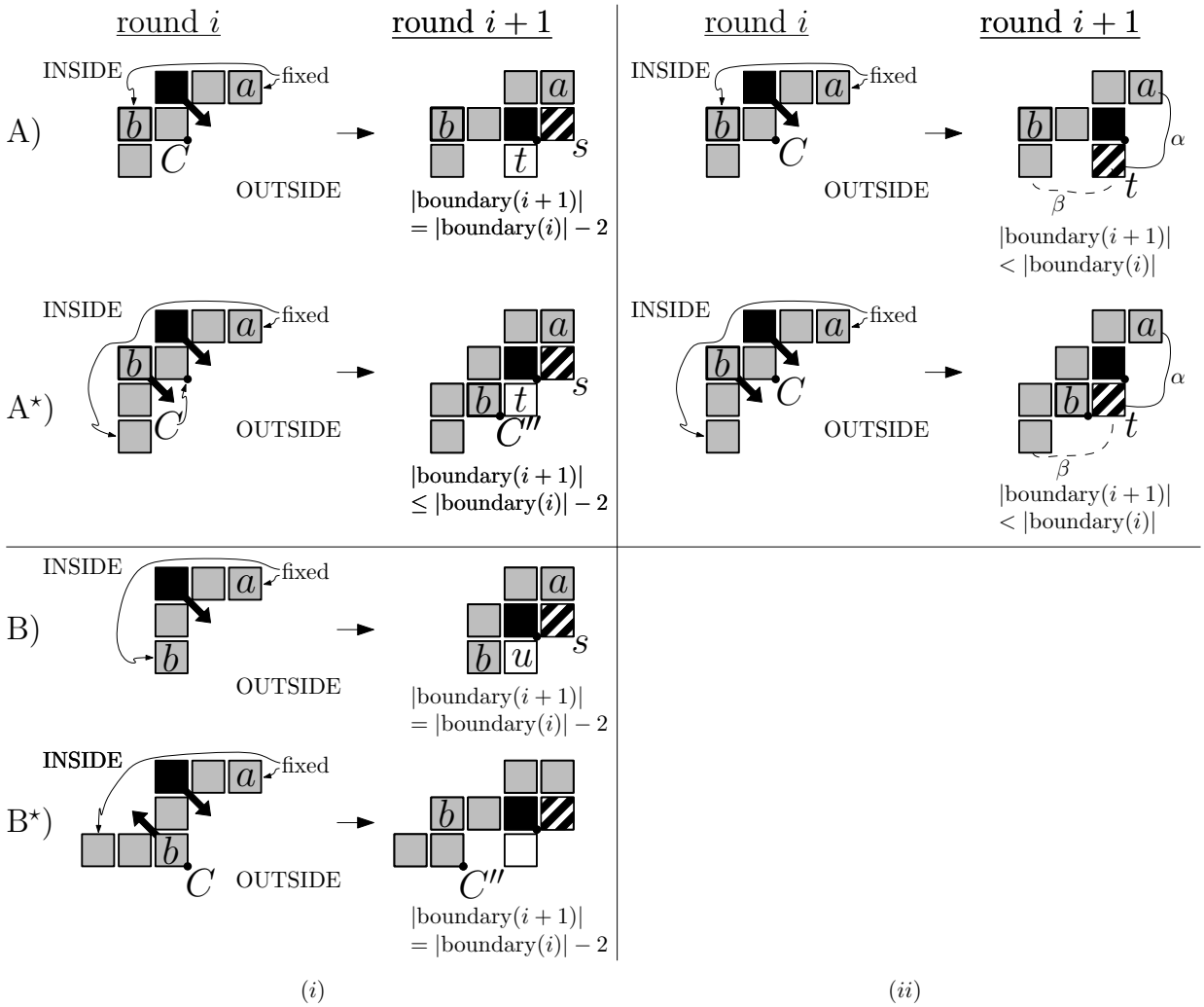

Figure 8: Diagonal hops can change the outer boundary's length.

In the figure, all robots (in (i) and (ii)) hop towards the swarm's outside. In column (i), just the cell s contains a robot. Then, we see, that in all cases the boundary becomes shorter by at least 2. In case of $\mathrm{A}^\star$, this can be even more than 2 for the case that after the hop the cell below b (in Figure 7.(ii), this is the cell u) also contains a robot.

Column (ii) show the cases where the cell t is occupied.

Because the swarm is connected, the hatched robot must be connected to the rest of the swarm

(Else, the swarm was disconnected before the hop.). This can be either via the subboundary α or β. In both cases, the hop shortens the outer boundary. This happens by forming inner bubbles (cf. Figure 9). This finishes the proof. □

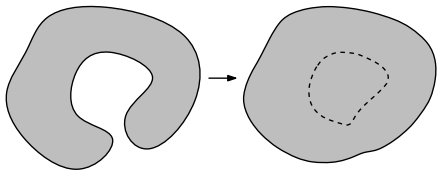

Figure 9: During the gathering, inner bubbles can be developed.

6.3 Progress measurement *Area*

The third progress measurement *Area* does not behave monotone. It can be reversed during rounds with *Boundary* or *Convex* progress. But we use it for estimating the number of the remaining rounds (Corollary 1). And in the proof of Theorem 1 we show that amount of the reversed progress does not have worsen the asymptotical running time.

Corollary 1. *If in a step of the gathering process neither* Boundary *nor* Convex *has progress, then instead* Area *has progress by at least* -8*.*

The tricky proof of Corollary 1 can be found in Subsection 7.1.

6.4 Total running time

Now, we can combine all three progress measurements *Boundary*, *Convex* and *Area*for the running time proof (Theorem 1).

Theorem 1. *A connected swarm of* n *robots on a grid, can be gathered in* $\mathcal{O}(|\text{outer boundary}|^2) \subseteq \mathcal{O}(n^2)$ *many rounds.*

Proof. Let B be the initial length of the swarm's outer boundary. We know from Lemma 1, that *Boundary* progress increases monotonously Then, progress in *Boundary* happens at most B times. By Lemma 1, *Convex* progress also increases monotonously. As every robot on the swarm's outer boundary can provide at most 4 convex vertices, *Convex* progress happens at most $4B$ times.

We estimate the rounds without *Boundary* and *Convex* progress via the size of the included area, i.e., the *Area* progress. By Corollary 1, we know that in every round without *Boundary* and *Convex* progress, the area becomes smaller by at least -8. But, *Area* is not a monotone progress measurement. Instead, in rounds with *Boundary* or *Convex* progress, the included area can increase:

The included area of the swarm is upper bounded by B^2. While HV hops cannot increase the included area, Diag-$\{$A, B$\}$ hops can. First, we assume, that the according hops do not change the outer boundary length. Then, every time *Convex* has progress, the area can become larger by at most B, because every robot hop on the outer boundary can increase the area by at most 1. As *Convex* happens at most $4B$ times, this in total is upper bounded by $4B^2$.

If the outer boundary length changes, then the included area can increase (cf. proof of Lemma 2 and Figure 9). Then, a *Boundary* progress by ℓ can also increase the included area by $\Delta A \leq \ell^2 \leq \ell B$. But as *Boundary* progress is not reversible, the sum of all these ΔA is upper bounded by B^2.

Summing up, during the whole process of the gathering, the area can be increased by at most $(4+1)B^2 = 5B^2$. Together with the initial area of at most B^2, *Area* progress happens at most $6B^2$.

Then, the gathering is done after at most $B + 4B + 6B^2$ rounds. This finished the proof. □

7 Proofs for area progress

7.1 Proof of Corollary 1

In this section, we proof the Corollary 1. Our proof splits in to parts: We distinguish swarms that do not contain so called *bridges* which are the black robots in Figure 16 and those that contain them.

In Lemma 4, we deliver the proof only for swarms like in Figure 10, i.e., without bridges. Here, the basic argument is, that because during a complete walk around the swarm's outer boundary, a total rotation by 360° is performed, the number of convex vertices is 4 bigger than the number of concave ones, because every such vertex performs a rotation by ±90°. From this basis, we first construct corners that match Diag-{A, B} hop patterns. And from corners we construct supercorners with the additional property that the hops of the patterns at their endpoints are not inhibited by inhibition patterns. I.e., the according robots actually perform their hops.

Lemma 5 generalizes this to swarms with bridges. This is not trivial, because we will show that bridges can hinder important hops.

What we mainly do is a transformation of the swarmstructure to a tree (Figure 16). Then, we estimate, depending on the number of bridges that are adjacent to a subswarm, how many hops that would reduce the included area can be hindered at most. By this, we notice that the included area of many subswarms increase. We show that the included area of a subswarm with only one bridge (such a subswarm is a leaf in our tree transformation) still decreases. The tree transformation then helps us to show that there always exist enough leafs for compensating the area increase of subswarms that have more bridges.

In the following, we assume, that no HV hops can be performed on the swarm's outer boundary.

Furthermore, we interpret the Inhibition patterns of our algorithm as follows: Assume, we have a black robot as in Figure 1 for that one of the Hop patterns matches. Then, the Inhibition patterns ensure, that it only hops, if the robot r' in Figure 2 does not also hop. I.e., it prevents that the black robot hops downwards, while r' hops upwards. We say, the robots *collide*. From a global point of view, the Inhibition patterns do the following: The black robot always assumes, it was located on the swarm's outer boundary and hops towards the swarm's inside. Then, if checking for collisions, it ignores all inside robots. Because of the locality, the black robot does neither know if it is located on the swarm's outer boundary, nor if it hops towards the inside. That's why we need to show Lemma 3.

Lemma 3. *Let ∂S be the outer boundary of some swarm S and S° its interior robots. We analyse Diag-{A, B} hops.*

1. *If $r \in \partial S$ hops towards the swarm's inside, it also does this on S.*
2. *If $r \in \partial S$ does not hop towards the swarm's outside, it also does not hop on S.*

The rough arguments in the proof are, that Hop patterns for hops towards the swarm's inside cannot be changed by inside robots, because for this the cells outside the swarm (the white marked cells in the Hop patterns) must be occupied by new robots. The other way round, for hops towards the swarm's outside, these cells are located inside the swarm, so that they *can* be occupied by new inside tobots. For the formal proof, see Subsection 7.2.

For the *Area* progress proofs, we need a more formal description of the swarm's outer boundary.

Definition 1 (quasi line)**.** *We define a sub-boundary, called a horizontal quasi line, as follows: Cf. Figure 10.*

1. *It consists only of horizontal subboundaries of length* ≥ 3 *that are connected by* stairways *of height* 1 *or* 2*.*

2. *It begins and ends with three horizontally aligned robots.*

In Figure 10, *stairways* are marked by black robots. They are alternating left and right turns. In the figure, their endpoints are marked by bicolored robots.

Stairways that do not match the quasi line Definition 1, connect two neighboring quasi lines that can be either both horizontal, both vertical or one horizontal and the other one vertical. Our algorithm performs Diag-{A, B} hops at these connection points. In Figure 10, dashed lines border the according patterns for *quasi line* 1.

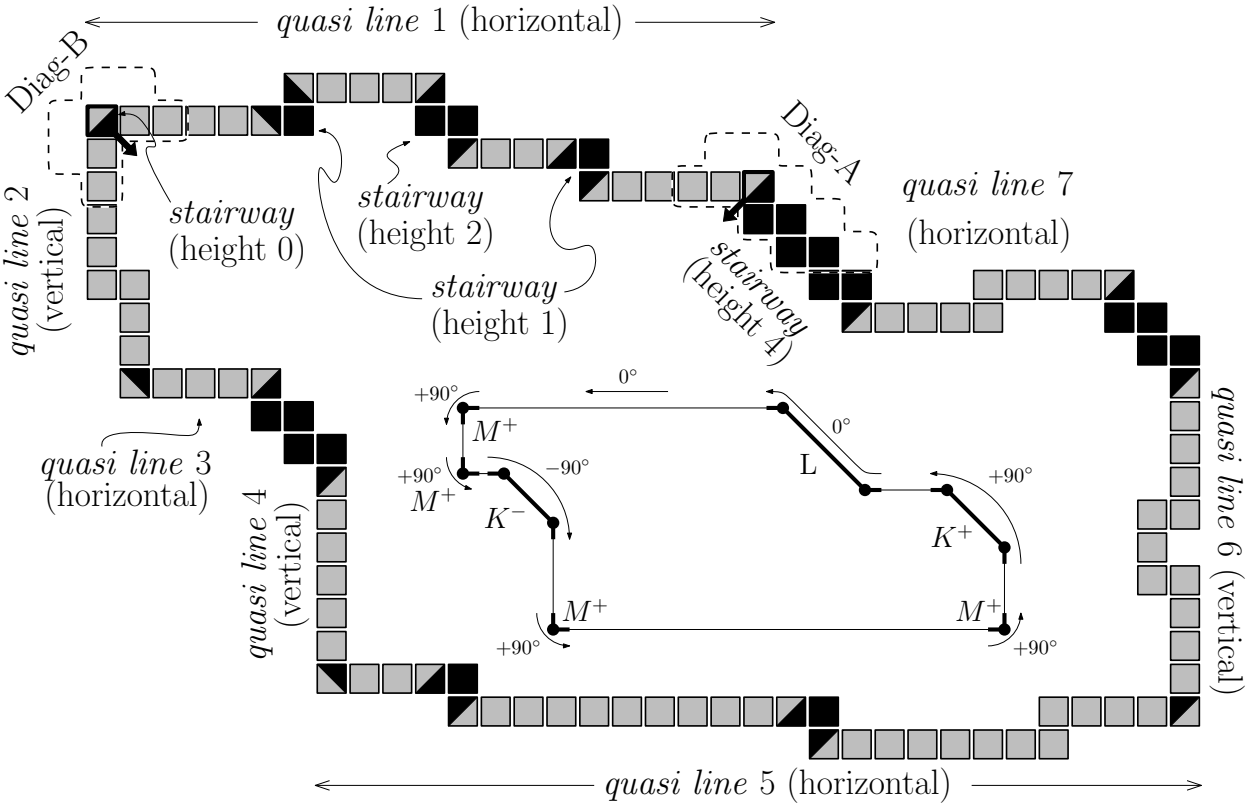

Figure 10: Example: Quasi lines and stairways. And a abstract representation of the shown boundary.

In our proofs, we represent a horizontal quasi line by a horizontal line segment and a stairway by a diagonal one. For the example in Figure 10, the drawn polygon shows this construction for the given example swarm. We call the connections between different quasi lines *corners*. These are the fat drawn parts of the polygon. At their endpoints, the Diag-{A, B} hops patterns match. The formal definition of *corners* is the following (cf. Figure 11).

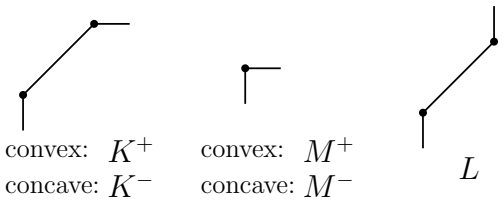

Figure 11: Definition of *corners*.

K^+ Corner includes a diagonal line segment and is convex (i.e. induces a $+90°$ rotation).
K^- Corner includes a diagonal line segment and is concave (i.e. induces a $-90°$ rotation).
L Corner includes a diagonal line segment and one end is convex and the other one concave (i.e. induces 0 rotation).
M^+ Corner does not include a diagonal line segment and is convex (i.e. induces a $+90°$ rotation).
M^- Corner does not include a diagonal line segment and is concave (i.e. induces a $-90°$ rotation).

Though the pattern matches one of Diag-{A, B}, a robot does not execute its hop, if the next robot(s) that hop in opposite direction, are too close, they *collide* (cf. Subsection 4.1). Figure 12

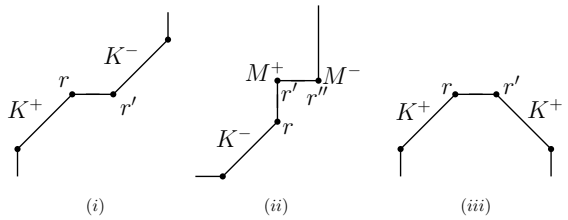

Figure 12: Robots that, according to the Diag-{A, B} patterns, would hop in opposite direction as one (Diag-A) or two (Diag-B) too close neighbor(s), do not perform this hop. (Significant examples)

shows significant examples for this: (i): r, r' do not hop. (ii): r, r' do not hop, but the robot r'' does, because only in one direction on the boundary is a robot that hops in opposite direction. (ii): $r,' r$ execute their hop, because they both hop downwards.

Now, we construct *supercorners* $\mathcal{X}, \mathcal{Y}, \mathcal{Z}$ from K^+, K^-, L, M^+, M^- corners, such that the supercorners have the additional property that only the two respectively the one (if it only consists of a $M^\pm$ corner) endpoint robots perform their hops. The hops of all other included K^+, K^-, L, M^+, M^- endpoint robots are hindered by *collisions* with neighbors. Because, in order to *collide*, the hops must be performed in opposite directions, the corners K, M along $\mathcal{X}, \mathcal{Y}, \mathcal{Z}$ must be alternating convex and concave. This means that we can define the supercorners analogue to corners (cf. Figure 13). Here, n_{K^+}, n_{M^+} denotes the number of K^+ respectively M^+ corners of the according supercorner and n_{K^-}, n_{M^-} denotes the same but for K^- and M^-, respectively.

$\mathcal{X}^+$: $n_{K^+} + n_{M^+} = n_{K^-} + n_{M^-} + 1$ (total $+90°$ rot.)
$\mathcal{X}^-$: $n_{K^+} + n_{M^+} = n_{K^-} + n_{M^-} - 1$ (total $-90°$ rot.)

$\mathcal{Y}$: $n_{K^+} + n_{M^+} = n_{K^-} + n_{M^-}$ (no total rotation)
$\mathcal{Z}^+$: $n_{M^+} = 1; n_{K^+}, n_{K^-}, n_{M^-} = 0$ ($+90°$ rotation)
$\mathcal{Z}^-$: $n_{M^-} = 1; n_{K^+}, n_{K^-}, n_{M^+} = 0$ ($-90°$ rotation)

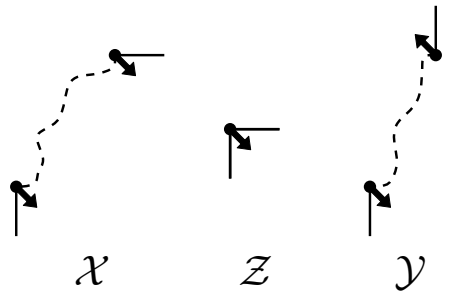

Figure 13: Definition of *supercorners* $\mathcal{X}, \mathcal{Y}, \mathcal{Z}$.

In the following lemmas, we distinguish two kinds of swarms: the ones with and the ones without *bridges*. For the definition of *bridges*, we first look at a swarm, that allows HV hops on the outside

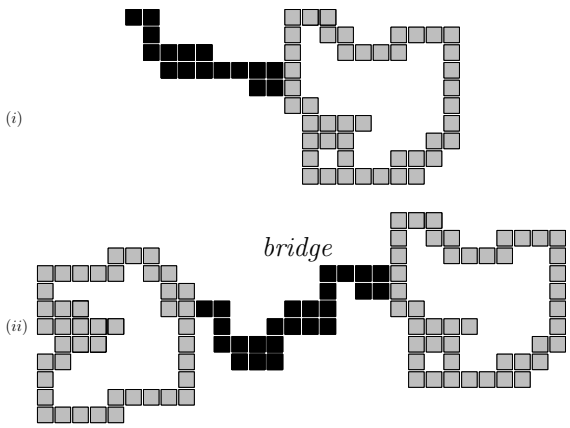

Figure 14: (i): The black subswarm can be completely removed only by executing a sequence of HV hops. (ii): The black *bridge* connects two subswarms.

of its outer boundary.

Such a swarm contains subswarms of widths ≤ 2 such that both of their sides border on the swarm's outside. They can be completely removed by executing a sequence that only contains HV hops (cf. Figure 14.(i)). If a subswarm B, of width ≤ 2 does not have an open end (in contrast to the one above), then it connects two wider subswarms. Then, we call B a *bridge* (cf. Figure 14.(ii)).

Lemma 4. *If a swarm $\mathcal{S}$ does not contain any bridges and neither* Boundary *nor* Convex *has progress, then instead* Area *has progress.*

Proof. During a walk along the outer boundary of the swarm, we perform a total rotation of 360°. Using the above construction, follows:

$$N_{K^+} + N_{M^+} = N_{K^-} + N_{M^-} + 4,$$

while $N_{K^+}, N_{M^+}, N_{K^-}, N_{M^-}$ are the total numbers of K^+, M^+, K^- and M^-, respectively. Then, there must exist neighboring corners without collisions. Then, by construction, the above equation also holds for supercorners:

$$N_{\mathcal{X}^+} + N_{\mathcal{Z}^+} = N_{\mathcal{X}^-} + N_{\mathcal{Z}^-} + 4$$

By the prerequisites of the lemma, there is no *Convex* progress. So, $N_{\mathcal{Z}^+} \stackrel{!}{=} 0$. Note, that $N_{\mathcal{Z}^-}$ can be bigger than zero, if the according hop is hindered by inside robots. We get:

$$N_{\mathcal{X}^+} = N_{\mathcal{X}^-} + N_{\mathcal{Z}^-} + 4$$

By construction, every every $\mathcal{X}^+$ executes two hops towards the swarm's inside, while $\mathcal{X}^-$ does the same towards the outside. Then, the included are behaves this way:

$$\Delta Area = -2(N_{\mathcal{X}^+} - N_{\mathcal{X}^-}) = -2(N_{\mathcal{Z}^-} + 4) \leq -8$$

This finishes the proof. □

Now, we generalize to a swarm that includes *bridges*.

Lemma 5. *If a swarm $\mathcal{S}$ contains $k > 0$ bridges and neither* Boundary *nor* Convex *has progress, then instead* Area *has progress.*

Proof. We first look at a subswarm S' that is connected by only one bridge B to the remaining part of $\mathcal{S}$. Later, we will call a subswarm with this property a *leaf*. For our analysis, we separate S' from $\mathcal{S}$, by splitting B into two parts (E.g., this could be done by removing ≤ 2 robots.). Afterwards, we remove the part that then is connected to S' by executing a sufficient number of HV hops. The remaining subswarm $\tilde{S}'$ from S' then only allows Diag-$\{\mathrm{A}, \mathrm{B}\}$ hops. Now, we start with the same construction as in the proof of Lemma 4, but this time applied only to the subswarm $\tilde{S}'$, Accordingly, we end up in the equation:

$$N^{\tilde{S}'}_{\mathcal{X}^+} + N^{\tilde{S}'}_{\mathcal{Z}^+} = N^{\tilde{S}'}_{\mathcal{X}^-} + N^{\tilde{S}'}_{\mathcal{Z}^-} + 4$$

In contrast to the proof of Lemma 4, here $N^{\tilde{S}'}_{\mathcal{Z}^+}$ can be bigger than 0, in case that B hinders the hops of all existing $\mathcal{Z}^+$ supercorners.

Now, we estimate the worst case for the number of diagonal hops towards the inside of $\tilde{S}'$ that could have been hindered by the bridge B. There are two cases how B can hinder a diagonal hop:

1. It occupies the white marked cells in the Diag-$\{\mathrm{A}, \mathrm{B}\}$ patterns.
2. It produces a collision with a robot that wants to perform a Diag-$\{\mathrm{A}, \mathrm{B}\}$ hop.

As the width of B is ≤ 2, it can hinder at most two hops towards the inside of $\tilde{S}'$. In Figure 15, we see that the $\mathcal{Z}^+$ supercorners ((i)) provide the worst case, because here we can hinder *all* hops of two convex $+90°$ supercorners. If, e.g., instead hops of $\mathcal{X}^+$ supercorners are hindered ((ii)), then afterwards still two of their robots hop towards the swarm's inside.

We get the equation

$$\begin{aligned} N^{\tilde{S}'}_{\mathcal{X}^+} + N^{\tilde{S}'}_{\mathcal{Z}^+} &= N^{\tilde{S}'}_{\mathcal{X}^-} + N^{\tilde{S}'}_{\mathcal{Z}^-} + 4 \\ N^{\tilde{S}'}_{\mathcal{X}^+} + 2 &= N^{\tilde{S}'}_{\mathcal{X}^-} + N^{\tilde{S}'}_{\mathcal{Z}^-} + 4 \\ N^{\tilde{S}'}_{\mathcal{X}^+} &= N^{\tilde{S}'}_{\mathcal{X}^-} + N^{\tilde{S}'}_{\mathcal{Z}^-} + 2. \end{aligned}$$

By construction, every $\mathcal{X}^+$ executes two hops towards the swarm's inside, while $\mathcal{X}^-$ does the same towards the outside. And as we, because of the prerequisites of the lemma, do not have *Convex*

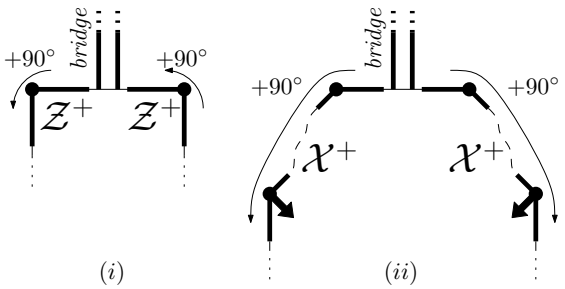

Figure 15: A bridge hinders at most two hops towards the swarm's inside.

progress, all $N^{\tilde{S}'}_{\mathcal{Z}^-}$ hops are hindered by inside robots. Then, the included area behaves as the following:

$$\Delta Area^{S'} = -2(N^{\tilde{S}'}_{\mathcal{X}^+} - N^{\tilde{S}'}_{\mathcal{X}^-}) = -2(N^{\tilde{S}'}_{\mathcal{Z}^-} + 2) \leq -4$$

I.e., in $\tilde{S}'$ respectively S', the included area becomes smaller by 4 instead 8 for swarms without bridges.

We can generalize this worst case construction to a subswarm $S^\star$ that is connected by $k \geq 1$ bridges to the remaining swarm. For this, we first analogously to the one-bridge-case, build $\tilde{S}^\star$ from $S^\star$ by removing the bridges.

There, still every bridge can hinder at most 2 hops towards the subswarm's inside and as before, the worst case means, that these must be $\mathcal{Z}^+$ corners. We get $N^{\tilde{S}^\star}_{\mathcal{Z}^+} = 2k$. So, in total

$$\begin{aligned} N^{\tilde{S}^\star}_{\mathcal{X}^+} + 2k &= N^{\tilde{S}^\star}_{\mathcal{X}^-} + N^{\tilde{S}^\star}_{\mathcal{Z}^-} + 4 \\ N^{\tilde{S}^\star}_{\mathcal{X}^+} &= N^{\tilde{S}^\star}_{\mathcal{X}^-} + N^{\tilde{S}^\star}_{\mathcal{Z}^-} + 4 - 2k \end{aligned}$$

and

$$\begin{aligned} \Delta Area^{S^\star} &= -2(N^{\tilde{S}^\star}_{\mathcal{X}^+} - N^{\tilde{S}^\star}_{\mathcal{X}^-}) \\ &= -2(N^{\tilde{S}^\star}_{\mathcal{Z}^-} + 4 - 2k) \\ &\overset{(*)}{\leq} -8 + 4k, \end{aligned}$$

where for $N^{\tilde{S}^\star}_{\mathcal{Z}^-} = 0$ $(*)$ becomes an equal sign.

This means, that for 2 bridges the area does not change and for > 2 bridges even increases. We will compensate this by showing that there are still enough subswarms that are only connected by a single bridge and call them *leafs*.

For showing this, we first transform the swarm to a tree (cf. Figure 16): We divide the swarm into bridges and subswarms S_i. In the graph representation, every S_i becomes a node while every bridge is interpreted as an edge. The resulting graph is a tree, because in our construction, an existing cycle would represent one of the subswarms S_i and so would contradict our construction in which every S_i is a node.

We estimate the total amount of area progress, by inductively constructing this tree: We start with the root node v_0 which has degree k_0. Then, also k_0 leafs are connected to this node. We know, that for this v_0 in the worst case the area behaves like $\Delta Area^{v_0} = -8 + 4k_0$. But adding the area change of the leafs, we get $\Delta Area \leq (-8 + 4k_0) - 4k_0 = -8$.

Now, we inductively add the other inner nodes to the tree. One can show that for every tree holds: Every inner node v of degree k serves $k - 2$ additional leafs. If we associate these leafs to

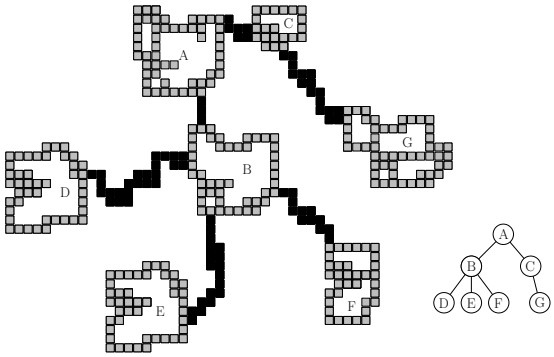

Figure 16: Transformation of a swarm with bridges to a tree.

v, we get $\Delta Area^v \leq (-8+4k) - 4(k-2) = 0$ So v doesn't have any impact on the total included area: $\Delta Area \leq -8+0 = -8$. This finishes the proof.

□

Lemma 4 and 5 immediately lead to Corollary 1.

7.2 Proof of Lemma 3

Proof. 1.) The hop of r can be hindered by inside robots by two reasons. a) The Hop pattern of r changes: This is impossible, as only inside robots can be added.

b) Collisions: (i): r changes from Diag-Bto Diag-A. Not possible because of a).

(ii): Some robot r' achieves hop pattern that makes him collide with r. Not possible because the Inhibition patterns ignore inside robots if r checks for collisions.

2.) Cf. Figure 17. The bold polygonal line marks the outer boundary. The hop of a r' that hops towards the swarm's can be enabled by inside robots by two reasons. a) The robot positions in the viewing range of r' changes: (i): r' changes from Diag-Ato Diag-B. Figure 17.(A): Not possible, because for this, s must be removed and t occupied. Both is not possible.

(ii): r' was located on a *quasi line* and changes to Diag-$\{\mathrm{A},\mathrm{B}\}$. α): Figure 17.(B): r' was located on a group of horizontally aligned robots. This would require to remove at least the robot t. β): Figure 17.(C): r' was located on a stairway of height 1. This would require to remove at least the robot s. γ): Figure 17.(D): r' was located at an endpoint of a stairway of height 2. This would require to remove one of the robots s, t. δ): r' could also not be located inside a stairway, because then the white marked cells in the Diag-$\{\mathrm{A},\mathrm{B}\}$ are not both free and also can't be emptied.

b) Collisions: (i): r' was Diag-A and becomes Diag-B. Not possible because of a).(i). (ii): Previously r' collided with some robot r and after adding inside robots, r does not collide anymore. Figure 17.(E): In the figure, OUTSIDE and INSIDE denote the real outside respectively inside of the swarm and $\mathrm{OUTSIDE}^{r'}$ and $\mathrm{INSIDE}^{r'}$ the way how r' assumes it, i.e., how the Inhibition patterns match, when checking for collisions.

The cells u, u' must stay empty because else the hop pattern for r' would be destroyed. When checking for hops, r' previously assumed $B^{r'}$ as the outer boundary.

α): Now, we could add a robot to s. Then r' assumes the outer boundary $B_1^{r'}$ and so r is interpreted as a Diag-B hop, which does not remove the collision. β): Instead, we can put robots to t, t'. Then, r' interprets $B_2^{r'}$ as the outer boundary, but then still a Diag-A hop pattern matches for r. γ): Figure 17.(F): Putting a robot on s would prevent r from performing a hop. But first,

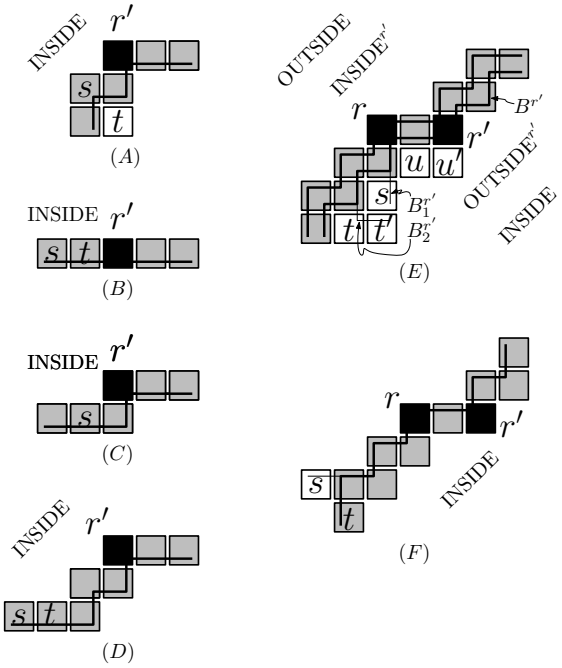

Figure 17: Illustrations for the proof of Lemma 3.

this is not in the inside of the swarm and second, the robot t must be removed which is also not allowed. □

References

[ACF+16] 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 *IPDPS '16*, pages 689–699, 2016.

[ASY95] Hideki Ando, Yoshinobu Suzuki, and Masafumi Yamashita. Formation and agreement problems for synchronous mobile robots with limited visibility. In *ISIC '95*, pages 453–460, August 1995.

[CFJM16] Andreas Cord-Landwehr, Matthias Fischer, Daniel Jung, and Friedhelm Meyer auf der Heide. Asymptotically optimal gathering on a grid. In *SPAA '16*, pages 301–312, 2016.

[CFPS03] Mark Cieliebak, Paola Flocchini, Giuseppe Prencipe, and Nicola Santoro. Solving the Robots Gathering Problem. In *ICALP '03*, pages 1181–1196, 2003.

[CP04] Reuven Cohen and David Peleg. Robot Convergence via Center-of-Gravity Algorithms. In *SIROCCO '04*, volume 3104 of *LNCS*, pages 79–88, 2004.

[DDSKN12] Gianlorenzo D'Angelo, Gabriele Di Stefano, Ralf Klasing, and Alfredo Navarra. Gathering of robots on anonymous grids without multiplicity detection. In *SIROCCO '12*, volume 7355 of *LNCS*, pages 327–338. 2012.

[DFKP06] Anders Dessmark, Pierre Fraigniaud, Dariusz R. Kowalski, and Andrzej Pelc. Deterministic Rendezvous in Graphs. *Algorithmica*, 46(1):69–96, 2006.

[DKL+11] Bastian Degener, Barbara Kempkes, Tobias Langner, Friedhelm Meyer auf der Heide, Peter Pietrzyk, and Roger Wattenhofer. A tight runtime bound for synchronous gathering of autonomous robots with limited visibility. In *SPAA '11*, pages 139–148, 2011.

[DKLMadH06] Miroslaw Dynia, Jaroslaw Kutylowski, Pawel Lorek, and Friedhelm Meyer auf der Heide. Maintaining communication between an explorer and a base station. In *IFIP TC10*, pages 137–146, 1 January 2006.

[DKM10] Bastian Degener, Barbara Kempkes, and Friedhelm Meyer auf der Heide. A local $O(n^2)$ gathering algorithm. In *SPAA '10*, pages 217–223, 2010.

[FPS12] Paola Flocchini, Giuseppe Prencipe, and Nicola Santoro. *Distributed Computing by Oblivious Mobile Robots*. Synthesis Lectures on Distributed Computing Theory. Morgan & Claypool, 2012.

[ISK+12] 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. *SICOMP*, 41(1):26–46, 2012.

[KM09] Jaroslaw Kutylowski and Friedhelm Meyer auf der Heide. Optimal strategies for maintaining a chain of relays between an explorer and a base camp. *TCS*, 410(36):3391–3405, 2009.

[KMP08] Ralf Klasing, Euripides Markou, and Andrzej Pelc. Gathering asynchronous oblivious mobile robots in a ring. *TCS*, 390(1):27–39, 2008.

[KTI+07] Y Katayama, Y Tomida, H Imazu, N Inuzuka, and Koichi Wada. Dynamic Compass Models and Gathering Algorithms for Autonomous Mobile Robots. In *SIROCCO '07*, volume 4474 of *LNCS*, pages 274–288, 2007.

[Mar09] Sonia Martînez. Practical multiagent rendezvous through modified circumcenter algorithms. *Automatica*, 45(9):2010–2017, 2009.

[Pre07] Giuseppe Prencipe. Impossibility of gathering by a set of autonomous mobile robots. *TCS*, 384(2-3):222–231, 2007.

[SN13] Gabriele Di Stefano and Alfredo Navarra. Optimal Gathering of Oblivious Robots in Anonymous Graphs. *LNCS*, 8179:213–224, 2013.

[SN14] Gabriele Di Stefano and Alfredo Navarra. Optimal Gathering on Infinite Grids. In *SSS '14*, pages 211–225. 2014.