

A Local $O(n^2)$ Gathering Algorithm*

Bastian Degener
Heinz Nixdorf Institute
Computer Science
Department
University of Paderborn
Germany
degener@upb.de

Barbara Kempkes
Heinz Nixdorf Institute
Computer Science
Department
University of Paderborn
Germany
barbaras@upb.de

Friedhelm Meyer auf der
Heide
Heinz Nixdorf Institute
Computer Science
Department
University of Paderborn
Germany
fmadh@upb.de

ABSTRACT

The gathering problem, where n autonomous robots with restricted capabilities are required to meet in a single point of the plane, is widely studied. We consider the case that robots are limited to see only robots within a bounded vicinity and present an algorithm achieving gathering in $\mathcal{O}(n^2)$ rounds in expectation. A round consists of a movement of all robots, in random order. All previous algorithms with a proven time bound assume global view on the configuration of all robots.

General Terms

Algorithms, Performance, Theory

Categories and Subject Descriptors

F.1.2 [Theory of Computation]: Modes of Computation; F.2.2 [Theory of Computation]: Nonnumerical Algorithms and Problems

Keywords

swarm robotics, local algorithms, distributed algorithms, gathering, geometric networks

1. INTRODUCTION

Over the last decade, there was a trend to consider large scale systems of autonomous robots with limited capabilities instead of systems with few but powerful computing entities. Such systems often are resilient against transient failures, scale well and behave well under dynamics. Yet it is not obvious which kinds of tasks robots with limited capabilities are able to solve. One challenge that gained great interest in recent years due to its simplicity is to gather a group of n robots in a common point. Most research focused on

*Partially supported by the EU within FP7-ICT-2007-1 under contract no. 215270 (FRONTS) and DFG-project "Smart Teams" within the SPP 1183 "Organic Computing" and International Graduate School "Dynamic Intelligent Systems"

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

SPAA'10, June 13–15, 2010, Thira, Santorini, Greece.
Copyright 2010 ACM 978-1-4503-0079-7/10/06 ...\$10.00.

the question of which robot capabilities are crucial in which time model to solve the gathering problem in finite time, often assuming the robots to have a global view on the current situation. Some work also exists about robots with a local view [10, 3, 2]. Here, the robots normally have a circular visibility range with a constant radius. However, apart from few exceptions, no runtime bounds are given. Considering locality is reasonable in large robotic systems to guarantee scalability. Moreover, in many practical applications global knowledge is not accessible to the robots. The main contribution of this paper is an algorithm that terminates in expected $\mathcal{O}(n^2)$ rounds, while our robots only have knowledge about their local environment from a constant visibility range.

Related work.

In the literature, several robot abilities are distinguished. When considering the gathering problem, robots are normally assumed to be *oblivious*, which means that the robots do not remember anything from the past. When taking a decision, they therefore rely only on the information which is available in the environment. Another common robot constraint is *anonymity*. In this case, robots do not know ID numbers and they cannot distinguish their neighbors from each other. Another type of constraint is the *compass-model*. If all robots use the same coordinate system, some tasks are easier to solve than if the robots' local coordinate systems can be arbitrarily distorted and scaled (the robots are called *disoriented*). There also exist some compass models in between. For example, the robots can share the directions of the coordinate system, but the axis-scales can be different.

Several time models were proposed. If no runtime bounds but convergence or termination is considered, asynchronous models are often used. In such models, robots may become active at any time, their algorithm executions may be split over several activation periods. In the most general case, the only restriction is that, on an infinite time scale, every robot is activated infinitely often. Time in such a setting is typically defined in terms of rounds. A round finishes as soon as every robot was active at least once. Such asynchronous models typically exclude the problem of concurrency, and therefore maybe interfering activations. This problem is crucial for many synchronous time models.

For the gathering problem, it was shown recently in [8] that robots which are anonymous, disoriented, oblivious and cannot communicate are able to gather in a so called semi-synchronous time model if and only if n is odd. In [18] and [12], the effect of compass models is studied under various aspects. Another focus are negative results, showing under which circumstances robots cannot gather (e.g. [17]). If at least one robot behaves maliciously, gathering is only possible if there are at least three robots in total [1]. In

[5], the authors not only restrict the robots by prohibiting communication, memory and a common coordinate system, but they also use robots which have an extent. The challenge here is that the view of a robot can be blocked by another robot. Again, the results are stated without runtime bounds. In [4], upper bounds of $\mathcal{O}(n^2)$ for the easier convergence problem in several time models are shown, but with robots having a global view. There are algorithms with local view and runtime statements for similar problems, such as transforming a long winding chain of robots into a short one [14, 9, 6]. There is also work for gathering on graphs instead of Euclidean spaces [7, 13, 15]. The authors in [11] point out that having no further assumptions on the robots leads to an exponential lower bound for randomized algorithms for gathering and propose a linear time gathering algorithm, on the base of multiplicity detection. However, in contrast to our work, their algorithm is not local and randomization is inherently needed. The results in [16], [3] and [2] are the closest to ours. In [16], robots converge to the convex hull of some stationary devices, [3] solves the robot convergence and [2] the robot gathering problem locally. All three algorithms basically let a robot move to the center of the smallest enclosing circle around its neighbors. It is shown that the algorithms solve the respective problems in finite time, but no runtime bounds are given.

Problem description and notation.

We are given a set N of n robots in the plane. We assume a discrete time model. A *configuration* at time t is described by the robot positions $p_1(t), \dots, p_n(t)$ at time t . The configuration at time 0 is called *start configuration*. If clear from the context, we will sometimes also refer to a robot r_i 's position by r_i . G_t denotes the unit disk graph on the positions. $d(r_i, r_j)$ describes the (Euclidean) distance between the two robots r_i and r_j ($d(r_i, r_j)$ varies over time, but it will always be clear from the context). Two robots are connected, if they are within distance 1 of each other. We call this distance the *connection range*. We assume G_0 to be connected. To make sure that the robots do not split into several groups, our algorithm will keep G_t connected at all times. In contrast, the *viewing range* defines the locality: Robots can see all robots within distance 2 and therefore twice as far as the connection range. (For our algorithm it would suffice if every robot knew the positions of all robots within its 2-hop neighborhood, but using the Euclidean distance for the viewing range simplifies the description.)

The goal is to gather the robots in one point, using the restricted robots as described in the robot model. In order to measure the quality of the algorithm, we count the number of rounds (see Round model) until they have gathered.

When a robot executes the algorithm, it computes *target positions* for itself and neighboring robots. These positions are reached before the next time step starts.

Our analysis is based on the convex hull of the robots. We distinguish the *global convex hull* $CH(t)$ at time t , which describes the convex hull of all robot positions at time t , and the *local convex hull* $C_r(t)$ of a robot r at time t . The local convex hull is the convex hull of all robots which are within viewing range of r at time t .

Round model.

One of the most common round models for robot formation problems is the asynchronous round model. Time is modeled as a sequence of discrete points of time (*time steps*). In each time step at most one robot is active. A *round* ends as soon as each robot has been active at least once. This model assumes that robots are never active concurrently, so no conflicts among these actions of active robots have to be handled. Usually, the analysis of robotic strate-

gies in this model is done assuming activation of robots in worst case order in each round.

In this paper we assume weaker models for activation: In the *random order model*, we assume that, in each step, a randomly, uniformly chosen robot becomes active. The choices in different steps are independent. Note that the expected number of steps per round is $\mathcal{O}(n \log n)$ in this model.

In the *random permutation model*, we assume a fixed random permutation of the robots to be initially chosen. This permutation then prescribes the order of activation in each round. Note that each round takes exactly n steps in this model.

These time models are used for the analysis of our algorithm. Implementations should be distributed and should allow parallel activations of robots. For example, a slight variant of the random order model can be implemented as follows: We assume synchronized time steps. In each step, each robot wakes up with some given probability p . An awoken robot becomes active, if no other robot in its connection range is awoken. Note that several robots may now be active concurrently. But as their connection ranges are disjoint, no interference between the actions initiated by the active robots will appear. Choosing $p = 1/n$ leads to a round model which is very close to the random activation model (up to a slightly non-uniform probability distribution, because a robot with few neighbors has a slightly larger probability for becoming active than one with many neighbors).

In Section 4, we will present a variant of this model which uses a probability for wake-up which is dependent on the number of neighbors in G_t . It employs a distributed protocol for handling interferences which is tailored to our gathering algorithm. We will prove a $\mathcal{O}(n^2)$ bound for the expected number of time steps instead of rounds in this model.

Robot model.

We use robots with limited capabilities, trying to understand which capabilities are necessary to prove runtime bounds for the gathering problem with local view. Our robots

- have a limited viewing range
- are oblivious (they do not have memory)
- do not have a common coordinate system
- can assign target positions to robots connected with them (the robots thus need to be able to communicate)
- can measure positions of robots within viewing distance relative to their own position accurately
- can compute complex geometric properties (straight lines through given points, angles, ...)
- can share a position with another robot.

The possibility to assign target positions to neighbors makes the robot model more powerful than the ones studied in the literature. On the other hand, we use the strong restriction of a local viewing range. Note further that communication cannot be used to perform complex tasks, since when using oblivious robots, all gathered information is lost after one step. Consequently, when active, a robot can

- see all positions of robots within its viewing range relative to its own position
- compute target positions for all robots within its connection range.

- tell robots within its connection range their target positions
- move to its own target position (In our strategy, the maximum travelled distance in one step is 2)

When inactive, a robot can

- be told a target position
- move to this target position (In our strategy, the maximum travelled distance in one step is 3)

As soon as two robots share the same position, our algorithm will keep the robots together. So we say that two robots fuse when they share the same position for the first time. Concerning the round model, they now act as one robot, that is, their probability to be activated in a time step t is equal to the probability of non-fused robots.

2. DESCRIPTION OF THE ALGORITHM

The main idea of the algorithm is as follows. Each robot that is a vertex of the convex hull of the robot positions within its local viewing range tries to decrease the area that is covered by the robots as much as possible under the constraint that the unit disk graph of the robots remains connected. In addition, if there are too many robots in a given area, the complexity of the problem will be reduced by fusing single robots into one. As soon as all robots are close together, they can gather in one final step. Note that robots assuming to be a vertex of the global convex hull of robots but which are only a vertex of their local convex hull do not do any harm, because they never leave the global convex hull of robots. Note further that since the robots have a limited viewing range, we must guarantee that the robots do not split into several groups which will never find each other again.

We can now formally describe the algorithm LOCALGATHERING. It is executed by robot r at the time t in which it is active.

See Figure 1 for an illustration of Step 3b of the algorithm. Note that the algorithm is deterministic. We will bound the expected value for the number of rounds until all robots have gathered in one point in the next section; the only randomness used is the stochastic round model. In particular, the algorithm can also be executed in an asynchronous worst case round model, the only difference is that we cannot guarantee the runtime in this case.

3. ANALYSIS OF THE ALGORITHM

This section is dedicated to the analysis of the correctness and runtime of the algorithm LOCALGATHERING. We will first show some preliminaries and then analyze the runtime, measured in the number of rounds needed until the robots have gathered. This number will be shown to be $\mathcal{O}(n^2)$ in expectation, where the randomness comes only from the stochastic round model, while the algorithm itself is deterministic.

Preliminaries.

In order to prove that the robots gather in one point, we first show that G_t stays connected at all times and thus that the robots do not split into several groups. We prove this in the following lemma.

LEMMA 3.1. *If the network is connected before a robot r executes the algorithm, it is still connected afterwards.*

LOCALGATHERING: The algorithm for robot r at time t :

1. Compute the sets A_r and B_r of the robots within the viewing resp. connection range of r . Let C_r denote the convex hull of A_r .
2. (Termination) If $A_r = B_r$ (i.e. no robots from A_r have distance between 1 and 2 to r), then move all robots from A_r to the position $p_r(r)$ of r .
3. Else (B_r is a proper subset of A_r)
 - 3.a (Fusion) If the positions in B_r can be rearranged such that the resulting new set A'_r is still contained in C_r , is still connected, and at least two robots share the same position (are fused), perform this rearrangement. Fused robots will always have the same position from now on.
 - 3.b (Reduction) If fusion is not possible and r lies on the boundary of C_r , they do the following:
 - (a) Compute the two first intersections of the boundary of C_r with the boundary of r 's connection range if started from $p_r(t)$ in clockwise/ counterclockwise direction. (Note that these are the intersections which are in maximum distance to each other.)
 - (b) Compute the line segment l between these intersections.
 - (c) Move all robots on r 's side of l to their respective closest point on l .

PROOF. If the action in step 2 of the algorithm is executed, no robot was in the viewing, but not the connection range of r . Therefore, the robots which are moved to $p_r(t)$ were only connected to robots which are moved to the same point, keeping the connection. If r executes the action in step 3a of the algorithm, the robots in r 's viewing range stay connected by definition. Moreover, since only robots within r 's connection range are moved, edges of the unit disk graph G_t ending outside r 's viewing range are not affected.

Now let r be a robot executing the action in step 3b of the algorithm in time step t . Since again only robots within r 's connection range are moved, we only need to prove that all robots in the local convex hull $C_r(t)$ stay connected. For these robots (we now denote them by R), the straight line s which contains l separates R in two disjoint subsets R_1 and R_2 (let R_1 contain the robots on l). See Figure 1 for an illustration. Let R_1 be the subset which contains r , and let v be an arbitrary robot from R_1 . According to the algorithm, all robots from R_1 are moved to their projection on l , if it exists, and otherwise to their closest point on l (which is the closer end of l). It follows that the distance of v to its neighbors in R_1 can only decrease. If v is moved to its projection on l , by the definition of a projection its distance to the robots in R_2 can also not increase. If no projection of v on l exists, the movement of v can be split in two: If v was moved to its projection v' on s , the distance to its neighbors in R_2 would also not grow. From v' , v can be moved to its target position by projecting it to another straight line s' : s' is orthogonal to l and intersects l in v 's target position (and thus in the end of l which is closer to v). Again, all robots from R_2 are positioned on the other side of s' from v 's point of view, and thus its distance to the robots from R_2 can again not increase. \square

COROLLARY 3.2. *If Step 2 of the algorithm (termination) is executed, the algorithm has gathered all robots in one position.*

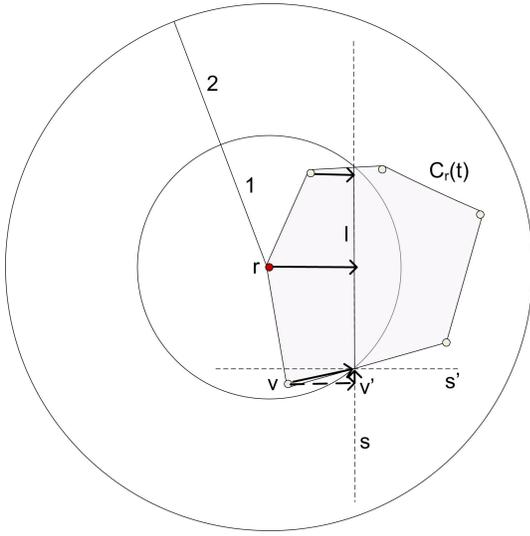


Figure 1: Illustration of step 3b of the algorithm and its correctness

Note that in rounds without fusions, if two robots were in each others connection range before the round, they still are afterwards.

In order to compute the number of rounds until the robots have gathered, we use two progress measures. Since fused robots never part again, fusing robots is progress. The other measure is the area of the convex hull which is truncated in one round. We will prove that we have progress concerning at least one of the two measures in each round: Either two robots are fused or the area decreases in expectation by a constant. Since fused robots never part again, the first measure is monotonically decreasing. We now show that this also holds for the second one.

LEMMA 3.3. *For all t and t' with $t' > t$ it holds: $CH(t') \subseteq CH(t)$.*

PROOF. If robots are rearranged while two robots fuse, a robot leaves neither the local convex hull nor the global convex hull of robots. If a vertex r of the convex hull moves itself and neighboring robots to the line segment l , l is completely inside the local convex hull of r and therefore again no robot leaves the global convex hull. \square

The next Lemma states another helpful fact and shows that a simple implementation of Step 3a of the algorithm is sufficient.

LEMMA 3.4. *If $|B_r| > 16$, then Step 3a of the algorithm (Fusion) is always possible.*

PROOF. Insert a grid with step width 1 into the intersection of r 's viewing range and $C_r(t)$. It is always possible to insert such a grid which has at most 16 points. If there are more robots in r 's connection range than points on the grid, moving the robots to the grid points guarantees that the unit disk graph of the robots in r 's viewing range stays connected and that no robot leaves $C_r(t)$. \square

Progress in rounds without fusions.

Since we start with n robots, there can be at most $n - 1$ rounds in which robots fuse. It remains to bound the number of rounds without fusions. In order to achieve this, we will prove that the area of the convex hull is decreased in expectation by a constant in

such rounds (Lemma 3.8). The idea of the proof is to bound the area which is truncated by a single robot which is a vertex of the global convex hull of robots (Proposition 3.6). This area directly depends on the internal angle of the global convex hull at the robot position at the moment the robot turns active. We show a relation between the internal angle at this moment and at the beginning of the round, so that we are able to sum up the progress of all robots by using the sum of the internal angles of the global convex hull at the beginning of the round.

Before we start with the proofs, we need to introduce some notation. In this subsection we will always consider a fixed round without a fusion. Moreover,

- m denotes the number of *vertex robots*, that is robots which are a vertex of the global convex hull CH at the beginning of the round. For ease of description, we renumber the vertex robots to r_1, \dots, r_m .
- β_i^* is the internal angle of the global convex hull at vertex robot r_i at the beginning of the round.

Now consider a vertex robot r_i which is still a vertex of the global convex hull $CH(t)$ in the first time step t in which it is active in this round.

- Let p_1 and p_2 denote the first intersections of the *global* convex hull $CH(t)$ with the boundary of r_i 's connection range if started from $p_{r_i}(t)$ in clockwise/ counterclockwise direction (the intersections which are in maximal distance to each other). Let T denote the triangle with the vertices $p_{r_i}(t)$, p_1 and p_2 . Then β_i is the internal angle of T in vertex $p_{r_i}(t)$.
- Let p'_1 and p'_2 denote the intersections of the *local* convex hull C_{r_i} and the boundary of r_i 's connection range if started from $p_{r_i}(t)$ in clockwise/ counterclockwise direction (the intersections which are in maximal distance to each other). Let T^* denote the triangle with the vertices $p_{r_i}(t)$, p'_1 and p'_2 . Then α_i is the internal angle of T^* in vertex $p_{r_i}(t)$.

Figure 2 illustrates the described angles. Note that $\alpha_i \leq \beta_i$, since the global convex hull contains the local convex hull at the beginning of time step t .

In order to bound the area which is truncated by a single robot, we start by showing that the internal angle of the local convex hull of this robot cannot be small, since otherwise robots can be fused.

PROPOSITION 3.5. *Consider a fixed round in which no robots are fused. Then α_i is greater than $\frac{\pi}{3}$ for all robots r_i which are a vertex of the global convex hull in the moment they turn active.*

PROOF. If $\alpha_i \leq \frac{\pi}{3}$ for a robot r_i , there exists one position p from which all robots are in distance at most 1 which were within viewing, but not connection range of $p_{r_i}(t)$. See Figure 3 for an illustration. r_i can be moved to the point of the local convex hull closest to the point shown as p in Figure 3. Moreover, because G_t is always connected, there must have been at least one robot in the connection range of r_i . All these robots can now fuse with r_i . Afterwards, no robots remain in the old connection range of r_i and thus the robots from $C_{r_i}(t)$ are connected. \square

PROPOSITION 3.6. *Consider a fixed round in which no robots are fused, and a robot r_i which is a vertex of the global convex hull in the time step t in which it turns active. The area of the global convex hull is reduced by at least $\frac{1}{2} \cos(\frac{\beta_i}{2})$ in this time step.*

PROOF. Consider the triangle T as defined above. Since the global convex hull $CH(t)$ contains the local convex hull $C_r(t)$, no

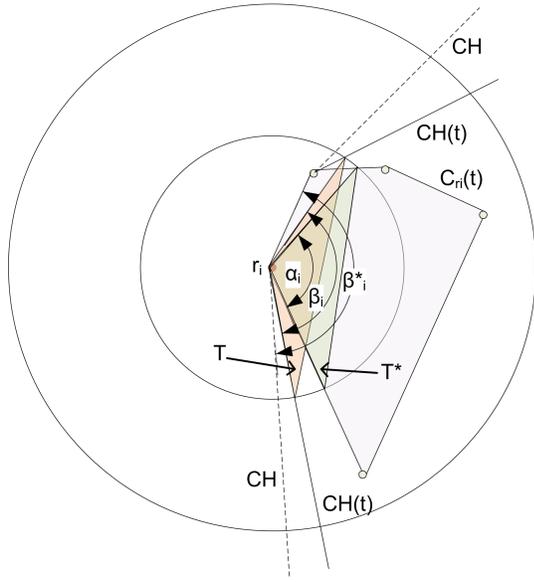


Figure 2: Angles used in this subsection. CH indicates the global convex hull at the beginning of the round. β_i and α_i are internal angles of the triangles T and T^* at the first time step in which r_i turns active in the round, β_i^* is the internal angle of the global convex hull at the beginning of the round.

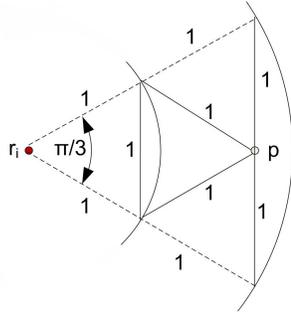


Figure 3: Illustration of a position from which all neighbors are in connection range. The indicated sector of the circle must contain the local convex hull, if $\alpha_i \leq \frac{\pi}{3}$.

point of l or the circular segment defined by l and the connection range of r_i can lie strictly inside of T (see Figure 4 for an illustration). As robot r_i moves all robots in its viewing range to this segment, the triangle T cannot contain any robots at the end of time step t . Since T is completely contained in the global convex hull, the area of the global convex hull is reduced by at least the area of T , which is $\sin(\frac{\beta_i}{2}) \cdot \cos(\frac{\beta_i}{2}) \geq \sin(\frac{\alpha_i}{2}) \cdot \cos(\frac{\beta_i}{2}) \geq \frac{1}{2} \cdot \cos(\frac{\beta_i}{2})$, where the first inequality follows from $\beta_i \geq \alpha_i$ and the second follows from Proposition 3.5: According to this proposition, α_i is at least $\frac{\pi}{3}$, giving that $\sin(\frac{\alpha_i}{2}) \geq \frac{1}{2}$. \square

The next lemma will be helpful when showing that the convex hull is reduced in expectation by a constant $\frac{1}{c}$. The constant c is the maximum number of robots that can be within the viewing range of a robot without fusing at least two of them. Lemma 3.4 states an upper bound for c of 16.

LEMMA 3.7. *A vertex robot r_i is never moved by the activation of another robot prior to its own activation with probability at least*

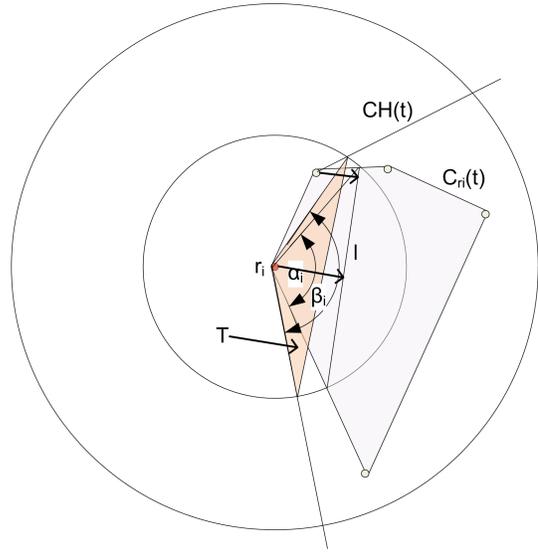


Figure 4: Illustration of the proof of Proposition 3.6

- a) $\frac{1}{c}$, in the random order model
- b) $\frac{1}{4c}$, in the random permutation model.

PROOF. At the moment t in which it is active, a robot can only have c neighbors (in its connection range), since otherwise robots could be fused. This also holds for all time steps of the current round before t , because once neighbors, robots stay connected at least until the next fusion. Therefore, the probability that a vertex robot r_i is active before any of its neighbors in the random order model is at least

$$\sum_{t=0}^{\infty} \frac{1}{n} (1 - \frac{c}{n})^t = \frac{1}{n} \frac{1}{1 - (1 - \frac{c}{n})} = \frac{1}{c}$$

In the random permutation model, the probability is at least

$$\frac{1}{n} + \left(\sum_{t=1}^{n-1} \frac{1}{n-t} \left(\prod_{j=0}^{t-1} 1 - \frac{c}{n-t} \right) \right)$$

Easy computations show that this is $\geq \frac{1}{4c}$. \square

LEMMA 3.8. *Consider a fixed round in which no robots are fused. The expected value for the area by which the global convex hull is reduced in this round is at least $\frac{1}{c}$ in the random order model resp. $\frac{1}{4c}$ in the random permutation model.*

PROOF. Let r_i denote a robot which is a vertex of the global convex hull CH at the beginning of the round. Remember that β_i^* is the internal angle of CH of robot r_i . Since CH is a convex polygon, the sum of its internal angles is $\sum_{i=1}^m \beta_i^* = \pi \cdot (m - 2)$. We want to use this sum of internal angles to determine the area by which the global convex hull is truncated in this round. Since robot r_i truncates the global convex hull by $\frac{1}{2} \cos(\frac{\beta_i}{2})$ (Proposition 3.6), we need a relation between the internal angle of CH in robot r_i in the beginning of the round (β_i^*) and β_i , the internal angle in robot r_i of the triangle T at the beginning of the first time step t in which r_i is active.

Note that r_i may have neighbors which were moved by other robots before. Consider the case that r_i is active before any other robot in its connection range. This means that r_i 's position cannot have changed in this round before time step t . If other robots have

moved before time step t in this round, they have not left the global convex hull and thus it can only have shrunk. This means that the internal angle of the global convex hull in robot r_i can only have decreased: At the beginning of time step t it is smaller than or equal to β_i^* . Finally, the triangle T is completely contained in the global convex hull at time step t . It follows that β_i is not larger than the internal angle of the global convex hull at r_i at the beginning of time step t , and thus $\beta_i^* \geq \beta_i$, if r_i is the first vertex of the convex hull which becomes active in its neighborhood.

Now we compute a lower bound for the expected area by which r_i truncates the global convex hull in the time step t in which it is first active, depending only on its internal angle of the global convex hull at the *beginning* of the round. For this, let a_i denote the random variable which describes the area truncated by robot r_i in the round. The following computations are for the random order model, they can be computed analogously for the random permutation model.

$$\begin{aligned} E[a_i] &\geq \Pr[r_i \text{ is the first activated robot in its connection range}] \cdot (\text{area truncated in this case}) \\ &= \frac{1}{c} \cdot \frac{1}{2} \cos\left(\frac{\beta_i}{2}\right) \\ &\geq \frac{1}{2c} \cos\left(\frac{\beta_i^*}{2}\right) \end{aligned}$$

The equality follows from Lemma 3.7. Finally, lower bounding the cosine in the interval $[0; \frac{\pi}{2}]$ by the straight line g with $g(x) = 1 - \frac{2}{\pi}x$, we can use the sum of all internal angles of the global convex hull ($\sum_{i=1}^m \beta_i^* = \pi \cdot (m-2)$) to estimate the expected truncated area in the round:

$$\begin{aligned} E\left[\sum_{i=1}^m a_i\right] &= \sum_{i=1}^m E[a_i] \\ &\geq \sum_{i=1}^m \frac{1}{2c} \cos\left(\frac{\beta_i^*}{2}\right) \\ &\geq \frac{1}{2c} \left(\sum_{i=1}^m -\frac{2}{\pi} \cdot \frac{\beta_i^*}{2} + 1 \right) \\ &= \frac{1}{2c} \left(m - \frac{1}{\pi} \sum_{i=1}^m \beta_i^* \right) \\ &= \frac{1}{2c} \left(m - \frac{1}{\pi} \cdot \pi \cdot (m-2) \right) = \frac{1}{2c} (m - m + 2) \\ &= \frac{1}{c} \end{aligned}$$

□

Note that we only use the first time a robot turns active in a round. If it turns active again, it may reduce the size of the convex hull further, but it will never increase the convex hull (Lemma 3.3). It follows that activating robots more than once in a round can only improve our result.

Runtime of the algorithm.

We can now put together the results to bound the number of rounds until the robots have gathered.

THEOREM 3.9. *Our local gathering algorithm needs expected $O(n^2)$ rounds in the random order and the random permutation model.*

PROOF. In each round, each robot r performs exactly one of the following three operations:

1. it moves all robots in its viewing range to its own position
2. it fuses two robots
3. if it is a vertex of the convex hull of its neighboring robots, it truncates a part of this local convex hull, otherwise it does nothing.

Since the unit disk graph of the robots always stays connected (Lemma 3.1), the first operation is only executed by a robot r if all robots are within connection range of r . It follows that after executing the first operation, the gathering has been achieved. Consequently, there is only one round in which this operation is performed.

The second operation can be performed at most $n-1$ times, since fused robots never part again and after at most $n-1$ fusions, all robots have fused to one robot.

The global convex hull of the start configuration can have an area of at most n^2 , because we assume the G_r to be connected. Since according to Lemma 3.8 the area of the convex hull is truncated in expectation by a constant in a round without fusions and since the area of the global convex hull never increases (Lemma 3.3), there can be at most $O(n^2)$ rounds in expectation without fusions. Summing up the number of rounds for each operation leads to the desired bound. □

4. VARIANT OF THE ROUND MODEL

So far, we have formulated our round models in a global fashion. Now we show that a variant also exists which can be implemented in a distributed synchronous setting.

Consider the following local activation protocol: In a time step, each robot r first computes the size b_r of B_r , the set of robots within its connection range. Then it wakes up with probability $\frac{1}{\max(c, b_r)}$, where c is the maximum number of neighbors a robot can have without fusing robots. It becomes active if no other robot r_1 in B_r with a smaller b_{r_1} woke up. If r is active, it performs our local gathering algorithm.

Note that the parallel executions never interfere. Note further that such a time step needs a computation of B_r , followed by just one step for parallel executions of our local gathering algorithm.

THEOREM 4.1. *Our local gathering algorithm needs expected $O(n^2)$ time steps in the local activation model.*

PROOF. Consider a time step.

If no fusion is possible, each robot r wakes up with probability $\frac{1}{c}$, and becomes active with probability $p \geq \frac{1}{c} \left(1 - \frac{1}{c}\right)^{c-1}$. As p is constant, Lemma 3.8 yields that, in such a time step, an expected constant size part of the convex hull is truncated. Thus, expected $O(n^2)$ time steps without fusion suffice.

If a fusion is possible, $b_r > c$ holds for some robot r . r wakes up with probability $\frac{1}{b_r} \left(1 - \frac{1}{b_r}\right)^{b_r} \geq \frac{1}{e \cdot n}$. Thus expected $O(n^2)$ time steps in which a fusion is possible suffice to perform all at most $n-1$ many fusions. □

5. CONCLUSION AND FUTURE WORK

We presented a local gathering algorithm with expected runtime $O(n^2)$. This is the first algorithm that solves the gathering problem with local view only and provides a runtime guarantee. We use a randomized round model in order to guarantee in our analysis the relation between inner angles of the convex hull at the beginning of a round and at the time in which a robot moves. This is a rather technical assumption and it would be worthwhile to study whether

it can be dropped in favor of an asynchronous worst case round model.

Furthermore, the runtime guarantee comes at the price of robots being able to move other robots within their vicinity. From an algorithmic point of view, the robots still act locally, for example in the sense that they do not know the global convex hull but rely only on the convex hull of robots in their vicinity. Still, this approach makes it difficult to apply the algorithm in a deterministic synchronous round model, or even in an asynchronous model where a robot can start to move while other robots have not reached their target position yet. It remains open if a similar runtime bound can be reached without this robot ability. In particular, having runtime bounds on the natural local algorithms considered in [16, 2, 3] that do not have this property would be nice. Our algorithm furthermore relies on the fact that the robots may look twice as far as needed for the connection of the network, which is not needed in [16, 2, 3]. Another open question is whether there are local gathering algorithms with runtime $o(n^2)$.

Note further that in order not to increase the convex hull globally, robots must not increase the convex hull locally. It would be nice to have lower bounds for any algorithm that obeys this constraint or at least for our algorithm.

6. REFERENCES

- [1] Noa Agmon and David Peleg. Fault-tolerant gathering algorithms for autonomous mobile robots. In *SODA '04: Proceedings of the fifteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 1070–1078, Philadelphia, PA, USA, 2004. Society for Industrial and Applied Mathematics.
- [2] Hideki Ando, Yoshinobu Oasa, Ichiro Suzuki, and Masafumi Yamashita. Distributed memoryless point convergence algorithm for mobile robots with limited visibility. *Robotics and Automation, IEEE Transactions on*, 15(5):818–828, Oct 1999.
- [3] Hideki Ando, Yoshinobu Suzuki, and Masafumi Yamashita. Formation agreement problems for synchronous mobile robots with limited visibility. In *Proc. IEEE Symp. of Intelligent Control*, pages 453–460, 1995.
- [4] Reuven Cohen and David Peleg. Convergence properties of the gravitational algorithm in asynchronous robot systems. *SIAM Journal on Computing*, 34(6):1516–1528, 2005.
- [5] Jurek Czyzowicz, Leszek Gasieniec, and Andrzej Pelc. Gathering few fat mobile robots in the plane. *Theoretical Computer Science*, 410(6-7):481 – 499, 2009. Principles of Distributed Systems.
- [6] Bastian Degener, Barbara Kempkes, Peter Kling, and Friedhelm Meyer auf der Heide. A continuous, local strategy for constructing a short chain of mobile robots. In *17th International Colloquium on Structural Information and Communication Complexity*, 2010.
- [7] Anders Dessmark, Pierre Fraigniaud, Dariusz R. Kowalski, and Andrzej Pelc. Deterministic rendezvous in graphs. *Algorithmica*, 46(1):69 – 96, 2006.
- [8] Yoann Dieudonné and Franck Petit. Self-stabilizing deterministic gathering. In *Algorithmic Aspects of Wireless Sensor Networks*, pages 230–241, 2009.
- [9] Mirosław Dynia, Jarosław Kutylowski, Paweł Lorek, and Friedhelm Meyer auf der Heide. Maintaining communication between an explorer and a base station. In *IFIP 19th World Computer Congress, TC10: 1st IFIP International Conference on Biologically Inspired Computing*, pages 137–146, 1 January 2006.
- [10] Paola Flocchini, Giuseppe Prencipe, Nicola Santoro, and Peter Widmayer. Gathering of asynchronous robots with limited visibility. *Theoretical Computer Science*, 337(1-3):147 – 168, 2005.
- [11] Taisuke Izumi, Tomoko Izumi, Sayaka Kamei, and Fukuhito Ooshita. Randomized gathering of mobile robots with local-multiplicity detection. In *Stabilization, Safety, and Security of Distributed Systems*, pages 384–398, 2009.
- [12] Taisuke Izumi, Yoshiaki Katayama, Nobuhiro Inuzuka, and Koichi Wada. Gathering autonomous mobile robots with dynamic compasses: An optimal result. In *Distributed Computing*, pages 298–312, 2007.
- [13] Ralf Klasing, Euripides Markou, and Andrzej Pelc. Gathering asynchronous oblivious mobile robots in a ring. *Theoretical Computer Science*, 390(1):27 – 39, 2008.
- [14] Jarosław Kutylowski and Friedhelm Meyer auf der Heide. Optimal strategies for maintaining a chain of relays between an explorer and a base camp. *Theoretical Computer Science*, 410(36):3391–3405, 2009.
- [15] Sonia Martínez. Practical multiagent rendezvous through modified circumcenter algorithms. *Automatica*, 45(9):2010 – 2017, 2009.
- [16] Friedhelm Meyer auf der Heide and Barbara Schneider. Local strategies for connecting stations by small robotic networks. In *IFIP International Federation for Information Processing, Volume 268; Biologically-Inspired Collaborative Computing*, pages 95–104. Springer Boston, September 2008.
- [17] Giuseppe Prencipe. Impossibility of gathering by a set of autonomous mobile robots. *Theoretical Computer Science*, 384(2-3):222 – 231, 2007. Structural Information and Communication Complexity (SIROCCO 2005).
- [18] Samia Souissi, Xavier Défago, and Masafumi Yamashita. Gathering asynchronous mobile robots with inaccurate compasses. In *Principles of Distributed Systems*, pages 333–349, 2006.