UNIVERSITÄT PADERBORN
Die Universität der Informationsgesellschaft

Local strategies for robot formation problems

Dissertation

by

Barbara Kempkes

Faculty of Computer Science, Electrical Engineering and Mathematics
Department of Computer Science and Heinz Nixdorf Institute
University of Paderborn, Germany

March 2012

Zusammenfassung

Wir betrachten eine Gruppe von mobilen, autonomen Robotern in einem ebenen Gelände. Es gibt keine zentrale Steuerung und die Roboter müssen sich selbst koordinieren. Zentrale Herausforderung dabei ist, dass jeder Roboter nur seine unmittelbare Nachbarschaft sieht und auch nur mit Robotern in seiner unmittelbaren Nachbarschaft kommunizieren kann. Daraus ergeben sich viele algorithmische Fragestellungen. In dieser Arbeit wird untersucht, unter welchen Voraussetzungen die Roboter sich auf einem Punkt versammeln bzw. eine Linie zwischen zwei festen Stationen bilden können. Dafür werden mehrere Roboter-Strategien in verschiedenen Bewegungsmodellen vorgestellt. Diese Strategien werden auf ihre Effizienz hin untersucht. Es werden obere und untere Schranken für die benötigte Anzahl Runden und die Bewegungsdistanz gezeigt. In einigen Fällen wird außerdem die benötigte Bewegungsdistanz mit derjenigen Bewegungsdistanz verglichen, die eine optimale globale Strategie auf der gleichen Instanz benötigen würde. So werden kompetititve Faktoren hergeleitet.

Reviewers:

- **Prof. Dr. Friedhelm Meyer auf der Heide, University of Paderborn, Germany**
- **Prof. Dr. Hans Kleine Büning, University of Paderborn, Germany**

Acknowledgements

I would like to thank my supervisor Friedhelm Meyer auf der Heide for his great support and for many motivating and fruitful discussions. Moreover, I would like to thank my colleagues, especially Bastian Degener and Peter Pietrzyk. I enjoyed working as well as spending time together. Last but not least, a big thank you goes to my husband Jens Peter, to my family and to my friends for their overall support.
Paderborn, March 2012 *Barbara Kempkes*

Contents

Chapter 1

Introduction

Envision a scenario, in which large teams of small and cheap mobile robots cooperate in order to perform global tasks. In this thesis, the considered task is to build a given formation out of an arbitrary configuration of initial positions. These kinds of tasks are called *robot formation problems*. It is especially interesting to figure out which sensor and actor capabilities are needed to build a given formation. Naturally, the goal is to require as few capabilities as possible in order to be able to use robots which are as cheap as possible. The focus of this thesis is especially on robots with a limited viewing range: robots can only "see" other robots within a circle with a fixed radius around their current positions. They have to decide where to move next only based on this incomplete information about the configuration of the whole team of robots. Nevertheless, the desired formation should be reached by the robots. This restriction of a local view is a very natural restriction for teams of small robots in large environments. Depending on the used technology, the viewing range of a robot can be larger or smaller, but no small and simple robot can look arbitrarily far. It is thus important to figure out which kinds of tasks can be solved by teams of such robots.

The goal of this thesis is thus to investigate how efficient robot formation problems can be solved by very simple robots with a local view: how far does the local view restrict the coordination of the robots? The considered problems are all easy to solve for robots with a global view and abilities as used in this thesis, but one single robot with a local view often cannot even decide whether the desired formation has already been reached. The challenge is therefore to coordinate the robots despite the very limited information, such that the desired formation is nevertheless achieved efficiently.

We introduce an approach to such formation problems that presents algorithms on a sufficiently abstract level, so that correctness and efficiency proofs are possible.

For this, we consider very simple models of robots and their environment: The environment is a plane without obstacles. The robots are considered as points in the plane, that is, they can neither block each other's view nor path. Besides the bounded viewing range, the robots have *no common compass*, and for many of our algorithms, the robots are *anonymous*. That is, they do not need to distinguish their neighbors. On the other hand, they can compute the exact relative positions of their neighbors within their viewing range, i.e. the distances to and the angles between the rays to these neighbors. Like most algorithms in the literature, we focus on *oblivious* algorithms, that is, the robots do not use a memory to remember the past. Moreover, most of our algorithms do not use communication. Thus a robot has to base its decision where to move next solely on the currently observed relative positions of its neighbors. If communication is used, the robots still cannot exchange global information, since the oblivious robots forget the gained information. It can only be used for local coordination.

The formation problems The formation problems considered in this thesis are the *robot chain problem* and the *gathering problem*.

The robot chain problem is defined as follows: In addition to n mobile robots $v_1, \ldots, v_n$, two stationary robots v_0 and v_{n+1} are given, which are called *stations*. We assume that, in the beginning, v_{i-1} and v_{i+1} are in the viewing range of v_i for $i = 1, ..., n$. Moreover, the decisions of v_i are only based on the relative positions of its direct neighbors v_{i-1} and v_{i+1}. Thus, the robots form a maybe winding chain connecting the two stations. The goal is to let all robots move towards the straight line between the two stations, the so-called *target line*.We will describe this problem and the underlying models in detail in Section 2.2.

The gathering problem is to let the n mobile robots gather in one point. This point is not prescribed, but the robots have to "agree" on a point based on their movement. No fixed neighborhood is given, but the robots may use the positions of all other robots within their viewing range, their neighbors, as basis for their decisions. In order to achieve gathering, we only require that the neighborhood graph, which has an edge between any pair of robots which are mutually visible, is connected at the beginning. A formal problem and model description can be found in Section 5.2.

These two problems capture two fundamentally different aspects of robot formation problems. Regarding the robot chain problem, the goal is to achieve a line as formation, which is not trivial. On the other hand, explicit neighborhood information is given to the robots, and the neighbors of a robot never change. This makes the analysis easier. In contrast, the gathering problem is simpler in the sense that the formation to be achieved is very simple, but then again, the neighbors of the robots are not fixed and can change over time. This complicates the problem.

The model For each formation problem, we will present several algorithms or *strategies*. All strategies adhere to the commonly used *Look-Compute-Move (LCM) model* [CP04]. That is, when a robot is active, it first observes its environment within its local viewing range (determines the relative positions of its neighbors), then it uses this information for computing a point towards which it wants to move (its *target point*) and finally it moves towards this previously computed target point. The algorithm or strategy for a formation problem defines the Compute-Operation: it takes the current positions of a robot's neighbors as input and outputs the target point.

The LCM model can be executed in several discrete and continuous, synchronous and asynchronous time and execution models. The algorithms presented in this thesis work in different of these time models. Therefore, the used time models will be introduced in detail together with the algorithms for both formation problems separately in the respective sections.

For a given strategy and a connected start configuration with n robots, we are first interested in the *correctness* of the strategy, that is, whether the strategy keeps the robot chain respectively the neighborhood graph connected. We are further interested in the *quality* of our algorithms. Typically, the quality of algorithms for robot formation problems is measured in terms of the *number of rounds*, where a round captures the time until each robot has been active (at least) once. The concrete definition depends on the time and execution model. We will also analyze the quality of our algorithms with respect to this quality measure. Additionally, we identify a second quality measure, which is the distance traveled by the robots. In particular, we analyze the maximum of the total distances traveled by the robots until the desired formation is reached, the maximum taken over the n robots. We will refer to this quality measure as the *(maximum) traveled distance* or the *movement distance*. This is a reasonable additional quality measure, since the distance traveled by the robots can vary a lot from round to round. Intuitively, it is preferable to keep the movement distance as small as possible. Moreover, energy is a major limiting factor for mobile robots. In terms of energy, the number of rounds reflects the number of neighborhood observations per robot. These can be expensive in terms of energy depending on the used hardware, for example if the robots have to stop moving before observing their surrounding. On the other hand, the movement distance of a robot reflects the energy which is needed for moving the robot.

Organization of the thesis After an overview over the related work, we will first investigate the robot chain problem (Chapters 2 to 4). We start in Chapter 2 with an introduction and a formal problem description, and we describe the used models and introduce our algorithms. Since the robot chain problem is well analyzed with respect to the number of rounds, we focus on strategies which are designed for reducing the maximum traveled distance (Chapter 3). For this, we introduce a continuous time and movement model, where the robots need to observe their neighborhood continuously and at all times. In this model, the notion of a round does not exist, and we only analyze the maximum traveled distance. Therefore, in Chapter 4, we combine both quality measures by using a discrete time model with a bound on the distance the robots may travel per round. We present one strategy for this scenario and analyze it with respect to both quality measures. We show that for the described strategy, no trade-off is required, but both the number of rounds and the maximum traveled distance can be minimized at the same time.

Chapters 5 to 8 are devoted to the gathering problem. We start again with an introduction, problem and model description (Chapter 5). Since to the best of our knowledge no analysis of the needed number of rounds is known for the gathering problem except for the ones presented in this thesis, we lay our main focus on the number of rounds. We present two algorithms, which both perform well regarding the number of rounds. The first one, which is described and analyzed in Chapter 6, constitutes a first algorithm with a known bound on the number of rounds needed. This is to the drawback that the robots need to be rather powerful, though still only local information is needed. The second algorithm (Chapter 7) compensates this drawback. The algorithm was already introduced in [ASY95, AOSY99, MS08] and it was shown that the robots gather, but so far no bounds on the time needed for gathering were known. We deliver this analysis and present a tight runtime bound. Finally, Chapter 8 considers gathering with respect to the maximum traveled distance. Similar to Chapter 3 for the robot chain problem, we use a continuous time model, and present an algorithm with very good bounds on the maximum traveled distance.

Chapter 9 concludes this thesis and raises open questions.

1.1 Related work

Several robot formation problems are considered from different perspectives in the literature. Examples for such problems, besides the robot chain problem and the gathering problem, are the convergence problem [SY99, CP05, CLDF$^+$11a, Kat11] and the circle formation problem [DK02, CMN04]. The convergence problem is a relaxed version of the gathering problem, with the difference that the robots do not need to reach a common point in finite time, but that they only need to converge to it. The goal of the circle formation problem is, as its name indicates, that the robots form a circle. A lot of effort has been put into pinpointing sets of needed robot capabilities in different time and movement models, such that the desired formation can be achieved. In many cases, a theoretical analysis or simulations are used to show that, with a specific algorithm and specific robot capabilities, the formation is reached, but statements about the quality of an algorithm are quite rare. Furthermore, there are several negative results, which show that some sets of capabilities are not sufficient to reach the formation. At this point, we want to highlight one publication which provides runtime bounds. Chazelle [Cha09] investigates the flocking problem, where a group of robots moves in some direction. Each robot realigns its movement direction and speed with those robots surrounding it. This problem has been investigated in several publications, and several algorithms and models have been proposed for this problem [JLM03, SWC05, Rey87], which show that the robots converge to the same movement direction and speed. Chazelle proves that, surprisingly, the convergence time is exponential for the most common models. Thus, it is especially interesting to analyze the time needed for robot formation problems in a local setting.

The robot capabilities which are needed to achieve a formation depend heavily on the time and execution model. Several such models were proposed. We will shortly describe the most common models here, but a detailed description of the models which we use in this thesis can be found in the respective sections. The synchronous model assumes that all robots are active and perform their algorithm concurrently [ASY95, DKLM06]. A relaxation is the semi-synchronous model, which was for example used in [DP09]. For this model, at a given point of time, a subset of the robots is activated. These robots perform the algorithm synchronously. Then a new subset of robots is activated. Asynchronous models are often used, if no runtime bounds but only termination is considered. There exist several variants of asynchronous execution models. A very general model (see for example [CP04, CFPS03]) assumes that the internal clocks of the robots are completely independent from each other. Robots may become active at any time, their algorithm executions may be split over several activation periods. The only restriction is that, on an infinite time scale, every robot is activated infinitely often. This fairness assumption is necessary, since if some robots stop being activated at some point of time, the robots have no chance of reaching the desired formation. Another asynchronous model activates the robots one at a time, such that no two robots are active concurrently. It was for example used in [MS08]. The execution of the algorithm cannot be split in several steps, but when a robot is active, it executes the complete algorithm. The order of activation can be determined by an adversary or randomly. 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 are often used to concentrate on the considered problem by excluding the issue of concurrency, and therefore maybe interfering activations. This problem is often crucial with a synchronous time model. In this case, symmetry breaking techniques have to be used in order to deal with neighbors interfering in the execution of the algorithm.

Due to their simplicity, a focus of the literature is on the *gathering* and the *convergence* problem. They have gained a lot of interest during the last 15 years. Many authors have studied robots which have a global view of the positions of the other robots [SY93, SY99]. Several articles have been published for the fully asynchronous setting, where the robots do not have a common notion of time. Cohen and Peleg showed that moving to the center of gravity of the robots leads to convergence, even in highly asynchronous models [CP04, CP05]. They also showed several runtime bounds which depend on the execution models. Some of these runtime bounds have been improved in [CLDF$^+$11a]. Furthermore, in [IIKO09] exponential lower bounds for the convergence of a certain class of randomized algorithms are shown. In [CFPS03], an algorithm was given that solves the gathering problem in a global and asynchronous setting, if the robots are able to detect whether there is more than one robot at a given point (multiplicity detection). Besides that, the robots are oblivious, they have no identities and no common coordinate system. In [DP09], it was shown that robots which are anonymous, do not have a common compass, are oblivious and cannot communicate, but which have a global view, are able to gather in the semi-synchronous model if and only if n is odd. In [SDY06] and [IKIW07], the effect of compass models was studied under various aspects. [Pre07] investigated situations in which the robots cannot gather. If at least one robot behaves maliciously, gathering is not possible in an asynchronous setting with three robots [AP04]. The authors also introduced an algorithm which gathers the robots in a synchronous setting if $n \geq 3f + 1$, where f denotes the number of faulty robots. In [CGP09], the authors did 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. [CLDF$^+$11b] also considered gathering of robots with an extent in a synchronous execution model. Here, the robots are not allowed to collide. The goal is to let the robots move as close together as possible. A runtime bound of $\mathcal{O}(nR)$ is stated for a grid terrain, with n being the number of robots and R the diameter of the start configuration. There is also work for gathering on graphs instead of

Euclidean spaces [DFKP06, KMP08, Mar09].

The gathering problem in the local setting was already tackled some time ago by Ando, Suzuki and Yamashita [ASY95]. Their robots move to the center of the smallest enclosing circle of their neighbors' locations. This target point definition guarantees that connectivity is maintained if no two robots are activated at the same time. But it can be easily seen that connectivity is not necessarily maintained in the synchronous setting. To overcome this problem, the authors restrict the distance that a robot moves towards its target point in a clever way, such that connectivity is guaranteed even under worst-case movement of the other robots performing the same algorithm. Furthermore, Ando, Suzuki and Yamashita showed that their algorithm allows the robots to gather in a finite number of rounds. Beyond this result, no runtime bounds were given. A follow-up article [AOSY99] evaluated the quality of their algorithm in a more realistic environment, where sensor data is not perfectly accurate, and suggested that the algorithm is robust against measurement errors of the sensors. The same algorithm, but in an asynchronous setting, was used by Meyer auf der Heide and Schneider in [MS08]. Here, the robots only move one at a time, and so the connectivity is maintained because the moving robots never moves out of sight of its neighbors. It is shown that the robots also gather in this setting. Furthermore, if some stationary nodes exist, the robots converge to the convex hull of these nodes. But again, no runtime bounds are given. We will analyze the runtime of the synchronous variant of this algorithm in Chapter 6.

Some further results exist for the gathering or convergence problem with a local viewing range. Flocchini, Prencipe, Santoro and Widmayer [FPSW05] showed that having a common orientation among the robots is sufficient to solve the gathering problem with a bounded viewing range in finite time in the fully asynchronous model. Katreniak [Kat11] also considered robots in the fully asynchronous setting with one restriction: During one LCM round of one robot, each other robot may start at most one of such rounds. That is, between two Look-Operations of one robot, each other robot may perform at most one Look-Operation. Besides having a limited viewing range, the robots are oblivious, do not have a common coordinate system, no identities and no communication. The presented algorithm solves the convergence problem in finite time. No runtime bounds are given.

The *robot chain problem* has also been considered before. In [DKLM06], Dynia, Kutylowski, Lorek and Meyer auf der Heide presented an intuitive strategy for a synchronous execution model: robots move synchronously to the old mid position of their neighbors. With this strategy, the robots converge to the line between the stations in $\Theta(n^2 \log n)$ rounds (see [DKLM06] for the upper and [KM11] for the lower bound). A second strategy was introduced by Dynia, Kutylowski, Meyer auf der Heide and Schrieb in [DKMS07]. This strategy achieves a linear runtime, but in exchange the robots need to know global coordinates as well as the position of one station. In [KM09], Kutylowski and Meyer auf der Heide introduced the more complicated and faster HOPPER-strategy. The idea is to let the robots hop over the midpoint between their two neighbors. Moreover, the strategy excludes robots from the chain if they are not needed for connectivity. This can also be seen as fusing two robots into one robot, a concept, which we will also use in this thesis.These operations combined with GO-TO-THE-MIDDLE-steps leads to a runtime of $\Theta(n)$ rounds until the sum of the distances between the robots is at most $\sqrt{2}$ times the distance between the stations. The strategy does not guarantee that the robots converge to the line between the stations, but its runtime is asymptotically optimal. For an overview of these strategies refer to the dissertation of Jaroslaw Kutylowski [Kut07].

A similar problem has been considered experimentally [NPR$^+$03, NPGF04, PNB07, STM10]. While one robot enters a building or becomes shielded by intervening terrain, a team of mobile relay robots maintains a communication chain to a base station. Some strategies for the relay robots have been tested on real robots. In [MS08], a local algorithm for a more general problem was considered: robots are distributed in the plane and have to shorten a communication network between several base stations. They have to base their decision on where to move in the next round on the relative position of all robots currently within distance 1.

1.2 Bibliography Note

Many of the results presented in this thesis have already been published in a preliminary version in conference proceedings. The analysis of the communication chain problem with respect to the movement distance (Chapter 3) has been first presented in
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 SIROCCO '10: Proceedings of the 17th International Colloquium on Structural Information and Communication Complexity, pages 168-182, 2010 [DKKM10].

Parts of the analysis for both quality measures (Chapter 4) were investigated in *Philipp Brandes, Bastian Degener, Barbara Kempkes, and Friedhelm Meyer auf der Heide. Energy-efficient strategies for building short chains of mobile robots locally. In SIROCCO '11: Proceedings of the 18th International Colloquium on Structural Information and Communication Complexity, pages 138-149, 2011* [BDKM11a].

An extended version of this paper has been invited for submission to a special issue of the Theoretical Computer Science Journal [BDKM11b]. Regarding the gathering problem, the algorithm presented in Chapter 6 was introduced and analyzed in *Bastian Degener, Barbara Kempkes, and Friedhelm Meyer auf der Heide. A local* $O(n^2)$ *gathering algorithm. In SPAA '10: Proceedings of the 22nd ACM symposium on parallelism in algorithms and architectures, pages 217-223, 2010* [DKM10].

The analysis of the algorithm which is presented in Chapter 7 was published in *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: Proceedings of the 23rd annual ACM symposium on parallel algorithms and architectures, pages 139-147, 2011* [DKL+11].

The results presented in Chapter 8 have not been published yet.

Chapter 2

The Robot Chain Problem

2.1 Introduction

We envision a scenario where two stationary devices (stations) and n mobile robots are placed in the plane. Each mobile robot has two neighbors (mobile robot or station) such that they form a chain between the stations, which can be arbitrarily winding and may intersect itself. Each robot has a limited viewing range of 1, that is, it can see its neighbors in the chain only if they are within distance 1 of itself. We assume that this is guaranteed in the beginning: Each pair of robots, which are neighbors in the chain, starts within distance 1 of each other. The goal is to design and analyze strategies for the mobile robots in order to minimize the length of the chain by moving the robots to the straight line between the stations (the *target line*), while always keeping the robots within viewing range of their neighbors. Each robot has to base its decision where to move solely on the current position of the neighbors in the chain— no global view, communication or long term memory is provided.

So far, the robot chain problem was analyzed in a synchronous setting with respect to the number of rounds needed until a straight line is almost achieved [DKLM06, DKMS07, KM09, KM11]. All strategies act according to the synchronous LCM computation model (Look-Compute-Move, see [CP04]). That is, for a given round t, first all robots observe their environment (determine the positions of their neighbors) at the same time, then they use this information for computing a point towards which they want to move and then all robots move towards this previously computed point at the same time. The robots only continue when all robots have finished their movement.

The simplest strategy, Go-To-The-Middle, was presented in [DKLM06]. Here, in each synchronous round, each robot moves to the middle between the old positions of its neighbors. It was shown that the robots converge to equally distributed points on the line between the stations, while they need $\Theta(n^2 \log n)$ rounds in the worst case until all robots are within distance 1 of the line (see [DKLM06] for the upper and [KM11] for the lower bound). As it does not make sense for the robots with a viewing range of 1 to move more than a distance of 1 per round, and as they can be at most in distance $n/2$ from the target line due to the chain being connected (the robots are in viewing range of each other), this strategy loses a factor of $\mathcal{O}(n \log n)$ rounds compared to an algorithm which knows the global situation.

Therefore, a second strategy, the Chase-Explorer-strategy, was presented in [DKMS07]. It is shown that the robots can converge to the line between the stations in linear time, if all robots know the position of the same station with respect to their own local coordinate system. But this is a clearly non-local information.

To overcome this lack of locality, the Hopper-strategy was introduced in [KM09]. This strategy does not let the robots converge to the target line, but it guarantees that the length of the chain (and the number of robots) is linear in the length of the target line after a linear number of rounds. For this, robots must have the ability to fuse with neighboring robots. This means that a robot moves to the position of one of its neighbors, and the two robots act like one robot from then on. The strategy is executed in sequential runs started at one station. A robot moves in the round after its predecessor has moved. These runs can be pipelined and are started in every third round. The idea of the strategy is to fuse a robot with one of its neighbors if its neighbors are within visibility range of each other, or otherwise to either move to or hop over the middle between its neighbors.

All of these strategies were analyzed with respect to the number of rounds. But, as described in the introduction, the movement distance is also a reasonable quality measure. To the best of our knowledge, the robot chain problem has not been considered with respect to the movement distance yet. The goal of this chapter is to tackle the robot chain problem integrating both quality measures. That is, we want to have strategies which keep both the number of rounds as well as the traveled distance small.

As Hopper does not let the robots converge to the target line, Go-To-The-Middle and Hopper are not comparable: There are configurations which cannot be improved by Hopper, but which still need $\Omega(n^2)$ rounds with Go-To-The-Middle until all robots are in distance 1 of the point to which they converge. In this thesis, we concentrate on strategies which are very simple with respect to the needed robot abilities, and which let the robots converge to the target line.

Outline As all known strategies were designed to minimize the number of rounds, our first question is what happens if we do it the other way round: We ignore the number of rounds by using a continuous time model and allowing the robots to observe their environment and the positions of their neighbors continuously and all the time. For this model, we design a strategy called Move-On-Bisector which we analyze thoroughly with respect to the movement distance (Chapter 3). As noted above, robots can be at most in distance $n/2$ from the target line. In the worst-case, strategies can therefore never be better than linear in n. We show that Move-On-Bisector reaches this bound, but we go even further: We show that Move-On-Bisector is better for configurations in which the robots already start close to the target line. In particular, we show that the maximum movement distance over all robots is upper bounded by $\mathcal{O}(min\{n, (OPT + d)\log(n)\})$, where OPT describes the maximum distance to be covered by robots with global view, and d the distance between the stations.

Then we turn towards the Go-To-The-Middle-strategy and the discrete and synchronous LCM model. The analysis for Go-To-The-Middle [DKLM06] already shows an upper bound of $O(n^2 \log n)$ for the number of rounds. Moreover, as the robots move at most a distance of 1 per round, the maximum movement distance is also upper bounded by $O(n^2 \log n)$. The strategy typically starts with large step sizes per round, while the step sizes become very small when the robots are close to the target line. We show that the maximum movement distance can nevertheless be $\Omega(n^2)$ and therefore a lot worse than that of Move-On-Bisector. But the movement distance can be improved if we change the strategy by bounding the step size of each robot by δ. We show that the movement distance decreases to $O(n)$ when using the continuous time model, but the main result is that for this strategy, we do not have to trade between the quality measures: By setting δ to $\Theta(1/n)$, the number of rounds worsens only by constant factors compared to the original strategy and stays at $\Theta(n^2 \log n)$, while the movement distance reduces to $\mathcal{O}(n)$.

Before we start with the analyses of the algorithms in Chapters 3 and 4, we now give details about the underlying model and the notation (Section 2.2), introduce the strategies (Section 2.3) and describe our quality measures in detail (Section 2.4).

2.2 Problem description and notation

We consider a set of $n + 2$ robots $v_0, v_1, \dots, v_{n+1}$ in the two-dimensional Euclidean plane $\mathbb{R}^2$. The robots v_0 and v_{n+1} are stationary and will be referred to as *base stations* or simply *stations*, while we can control the movement of the remaining n robots $v_1, v_2, \dots, v_n$. In the beginning, the robots form a chain, where each robot v_i is neighbor of the robots v_{i-1} (its *left* neighbor) and v_{i+1} (its *right* neighbor).

The chain may be arbitrarily winding in the beginning. The goal is to optimize the length of the robot chain in a distributed way, where the length refers to the sum of the distances between neighboring robots. That is, we want to let the robots move to or converge to the line between the stations (the *target line*), such that each robot is positioned between its left and its right neighbor. We are constrained in that the robots have a limited viewing range, which we set to 1. The robot chain is therefore *connected* if and only if for each two neighbors in the chain, the distance between them is less than or equal to 1. We assume that the chain is connected in the beginning. We say that a strategy for the robots is *valid* if it keeps the chain connected.

We want the robots to be as simple as possible in the sense that they do not need to have many abilities. In particular, our robots use only information from the current point of time (they are oblivious), share no common sense of direction and communicate only by observing the positions of their two neighbors. These observations are bounded to a local viewing range with radius 1 around the position of a robot. However, we require that the robots are able to distinguish their neighbors from the remaining robots in the robot chain (it is not necessary to distinguish the two neighbors from each other). On the other hand, we abstract from technical issues. In particular, we assume the robots to be able to measure positions of neighbors relative to their own position accurately, they can compute geometric properties and they can occupy the same position as other robots.

Time and movement models All our time models are based on the discrete LCM model as described in the introduction (Section 2.1). The classical model (see for example [CP04]) is as follows. During one *round*, all robots first observe the positions of their two neighbors relative to their own position (Look-Operation). Then they compute a *target point* (Compute-Operation), where the algorithm which determines the target point only gets the current positions of the two neighbors as input. Finally, all robots move to their previously computed target point (Move-Operation). The round ends as soon as all robots have reached their target point.

In order to reduce the movement distance, the idea of our strategies is to increase the number of Look-Operations and therefore the number of rounds. In order to do so, in Chapter 4 we restrict the robots performing GO-TO-THE-MIDDLE to move at most a distance of δ during one round. If a target point is in distance more than δ from the robot, it moves a distance of exactly δ towards its target point. Otherwise, the robot stops its movement as soon as it has reached its target point.

In Chapter 3, we introduce a strategy for which the robots need to observe their neighbors continuously at all times. The underlying model can be viewed as the

LCM model with a movement restriction to δ per round with $\delta \to 0$ and a speed limit of 1, yielding a continuous time model. That is, time passes in a continuous way and is not modeled by discrete time steps. Robots continuously observe the positions of their neighbors and adjust their trajectory and speed accordingly. As the robots only move in direction of their target point, the target point can be exchanged by a target direction. That is, at a fixed time t, a robot observes the positions of its neighbors and computes a direction in which it wants to move. Both operations need zero time. Then it moves in the computed direction. As a result, robots can move in curves and (by the definition of the time model) could even change the movement direction in a non-continuous way. We will see that this is not the case for the Move-On-Bisector-strategy. Here, the movement direction changes continuously, although speed adjustment can also occur in a non-continuous way. For this continuous time model, we can analyze the traveled distance, but rounds do not exist in this model. When seeing it as LCM model with $\delta \to 0$, one can also interpret each point in time as one round. From this point of view, the number of rounds in this model is infinite.

Notation Given a time $t \geq 0$ or a round $t \in \mathbb{N}_0$, the position of robot v_i at this time is denoted by $v_i(t) \in \mathbb{R}^2$. If not stated otherwise, we will assume $v_0(0) = (0,0)$ and $v_{n+1}(0) = (d,0)$, $d \in \mathbb{R}_{\geq 0}$ denoting the distance between the two base stations. To refer to the x- or y- coordinate of robot v_i at the end of a specific round or at a specific time t, we will use $x_i(t)$ and $y_i(t)$ respectively. We say that a robot v_i is in k hops from a robot v_j if $|i-j| = k$. A fixed placement of the robots (their positions at the end of round t) is called a *configuration*. In the discrete model, robots move during one round. While the configuration at time 0 is called the *start configuration*, robots move for the first time in the first round. The configuration at time 1 is therefore the configuration after the first movement. We call $d_i(t)$, $t \in \mathbb{N}_{\geq 1}$, the distance v_i travels in round t.

The vector connecting two neighboring robots v_{i-1} and v_i will be denoted by $w_i(t) := v_i(t) - v_{i-1}(t)$ for $i = 1, 2, \dots, n+1$. Moreover, let $\alpha_i(t) \geq 0$ be the smaller of the two angles formed by the vectors $-w_i(t)$ and $w_{i+1}(t)$. We will furthermore denote the scalar product of two vectors a and b simply by $a \cdot b$ and the length of a vector a by $||a||$.

Furthermore, we define two properties for a given configuration:

Definition 2.1 (height)**.** *The* height $\mathfrak{h}(t)$ *of a configuration at time t is the maximum distance between a robot in time t and the target line between the stations.*

Definition 2.2 (length)**.** *The* length $\mathfrak{l}(t)$ *of a configuration at time t is defined as the sum of the distances between neighboring robots:* $\mathfrak{l}(t) := \sum_{i=1}^{n+1} ||w_i(t)||$.

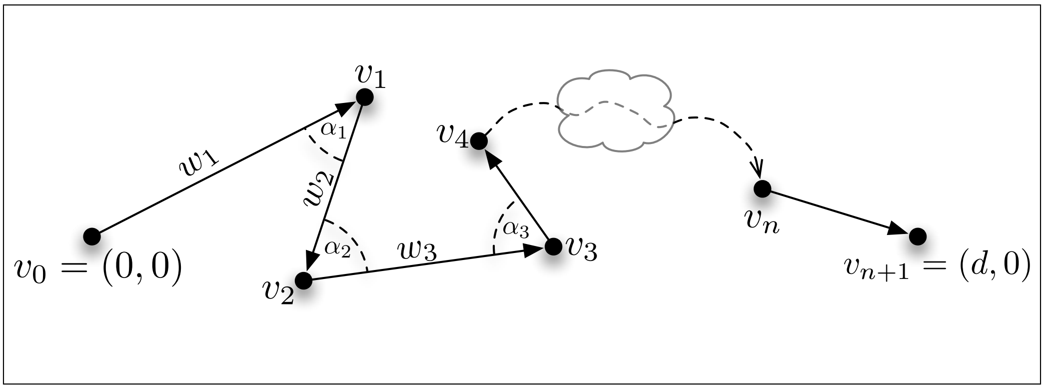


Figure 2.1: Notation for the robot chain problem. For clarity we omitted the time parameter t.

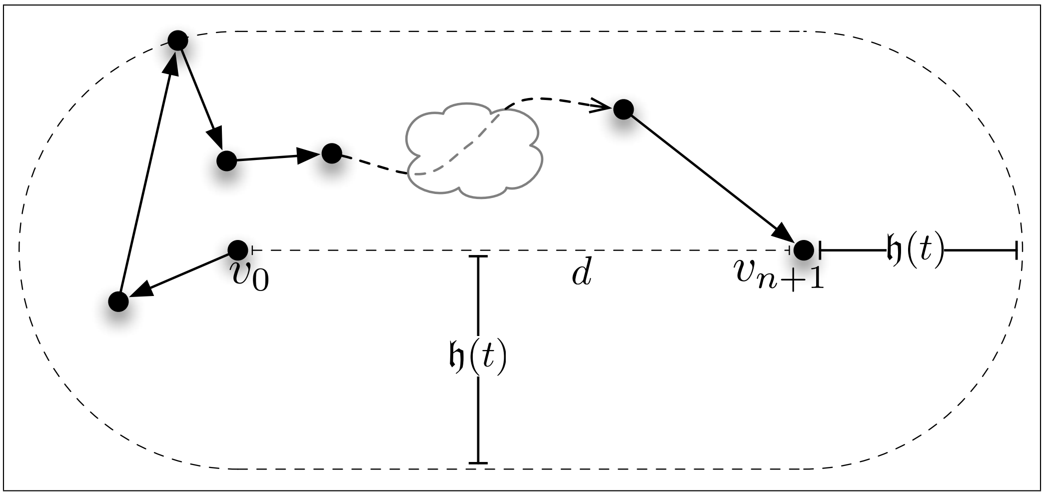


Figure 2.2: Illustration of the height $\mathfrak{h}(t)$ of a configuration.

Let $l := \mathfrak{l}(0)$ and $h := \mathfrak{h}(0)$. Clearly, the starting height h is a lower bound for the distance to be covered by an optimal global algorithm. The length of a configuration is also a natural quantity to measure its quality: a winding chain is relatively long compared to a straight line. Since the distance between two robots may be at most 1, it holds that $h \le \frac{l}{2}$ and $l \le n+1$. See Figure 2.1 and Figure 2.2 for an illustration of the notions defined in this section.

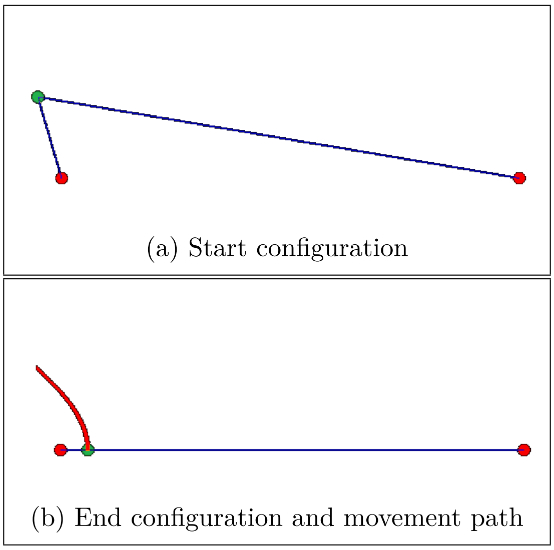
For the discrete model, we define $d_i(t) := ||v_i(t) - v_i(t-1)||$ to be the distance covered by robot v_i in the t-th round. Furthermore we set $d_i := \sum_{t=1}^{\infty} d_i(t)$ to the overall distance traveled by robot v_i.

2.3 Strategies

Now we have all preliminaries to describe all strategies formally. We start with the MOVE-ON-BISECTOR strategy, which works in the continuous time model and will be analyzed in Chapter 3. Then we introduce variants of the GO-TO-THE-MIDDLE strategy, which will be used to improve the movement distance of GO-TO-THE-MIDDLE in Chapter 4.

2.3.1 The Move-On-Bisector strategy

The MOVE-ON-BISECTOR strategy works in the continuous time model. That is, for each point of time t and a robot v_i, the strategy determines a movement direction and speed, given as the velocity vector of robot v_i. The goal is to see how far we can reduce the maximum traveled distance when we allow the robots to observe their neighborhood continuously and all the time.

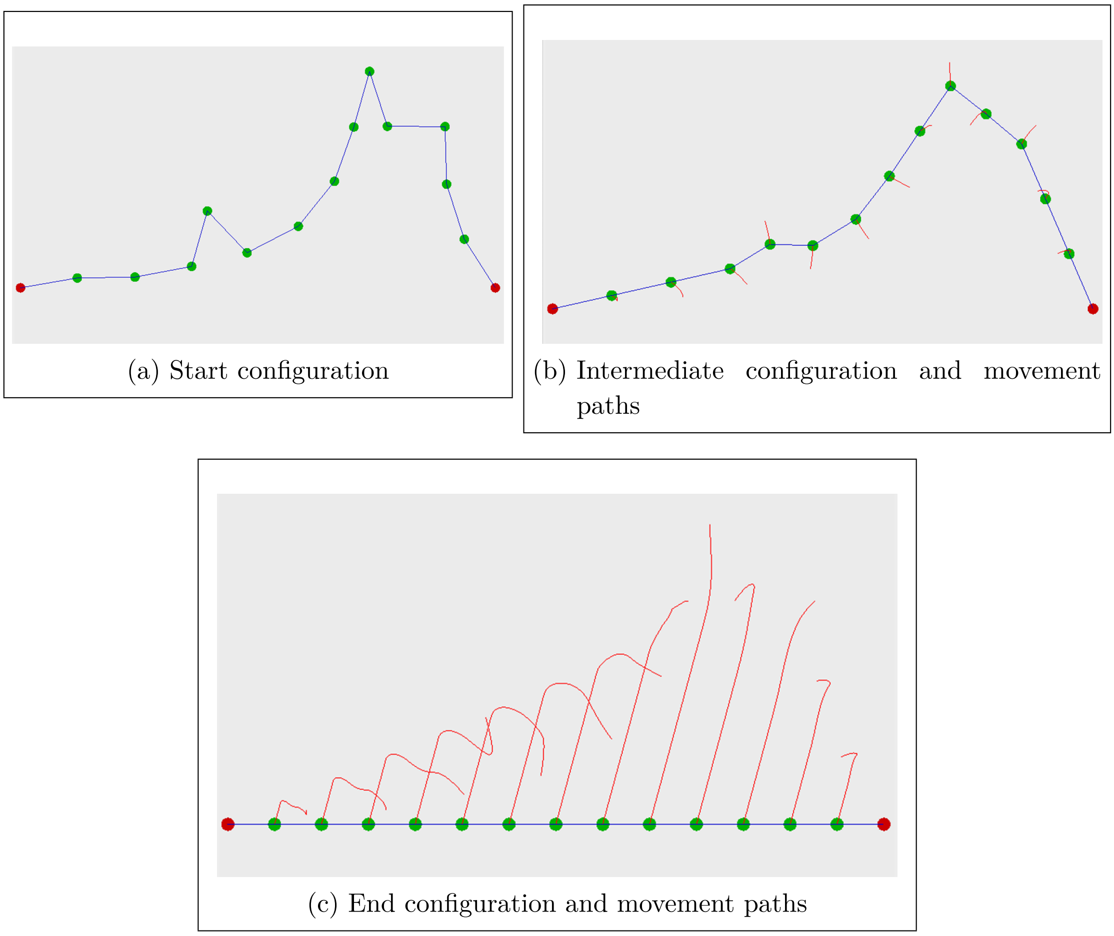


(a) Start configuration

(b) End configuration and movement path

Figure 2.3: Example for MOVE-ON-BISECTOR with one robot

The strategy works as follows. First consider those robots that have not yet reached the straight line between their two neighbors. We say that those robots are in *phase 1*. They move with maximum speed 1 in direction of the bisector of the angle α_i formed by the vectors pointing towards their two neighbors. As soon as a robot reaches the line between its neighbors, it adapts its velocity to stay on this (moving) line keeping the ratio between the distances to its neighbors constant (we say that the robot is in *phase 2*). Since the neighbors are also restricted to the maximum speed of 1, this is always possible: a robot will not have to move faster



(a) Start configuration

(b) Intermediate configuration and movement paths

(c) End configuration and movement paths

Figure 2.4: Example for MOVE-ON-BISECTOR with several robots which start in equal distances from their neighbors

than with speed 1 to stay on this line and to keep the ratio. Note that a robot never leaves the line between its neighbors again, it will therefore stay in phase 2 forever. Note further that with this strategy, the robots actually reach the line between their neighbors. Equally, they reach a point on the target line between the stations and do not only converge to it. Once all robots have reached such a point, they stop moving. We call this point the *end position* of a robot v_i. Moreover, we call the time when the last robot reaches its end position the *finishing time*.

One special situation can occur: If two neighboring robots v_i and v_j are at the same position at the same time, both take the other neighbor of v_j or v_i respectively as their new neighbor. Then, both robots have the same neighbors and will stay together from now on. This can be seen as two robots fusing into one robot, a concept, which we will use for the analysis of our gathering algorithms in Chapters 6 and 7. For the sake of clarity, we will ignore such situations in the analysis of MOVE-

On-Bisector.

Since robots which have reached the second phase stay in this phase until the end, all robots have reached the final line between the two stations as soon as all robots are in the second phase. Thus, the last robot reaching the second phase always moves with maximum speed 1. It follows that the maximum distance traveled by a robot is equal to the time until all robots have reached the target line.

2.3.2 The Go-To-The-Middle strategy

Now we describe the three variants of the Go-To-The-Middle-strategy ([DKLM06]), which we will analyze in Chapter 4. 1-GTM is the original Go-To-The-Middle-strategy, which moves all robots close to the target line in a reasonable number of rounds. On the other hand, we will show that the traveled distance is rather high. Continuous-GTM, a continuous variant of Go-To-The-Middle, is the other extreme: the number of rounds is not defined, but we will show that the traveled distance is small. δ-GTM, finally, is to take into account both quality measures: the traveled distance and the number of rounds.

δ-GTM We use the synchronous LCM model with rounds $t \in \mathbb{N}_0$. In each round, each robot computes the midpoint between its two neighbors. We will call this point the robot's *target point*. The robot moves towards its target point. However, we bound the distance the robots cover in one round to $\delta \in (0, 1]$. This implies that a robot v_i will reach its target point only if it is within distance δ of its own position. If this is not the case, the robot will move exactly a distance of δ towards it.

1-GTM This strategy is the original GTM-strategy from [DKLM06] and a special case of δ-GTM with $\delta = 1$. The robots always reach their target points. This strategy has already been intensively studied with respect to the number of rounds (for an overview, see [Kut07]), but the traveled distance has not been investigated before.

Continuous-GTM This strategy works in the continuous time model. As long as a robot v_i has not yet reached its target point, which is again the midpoint between its neighbors, it moves towards it with velocity 1. Once it has reached its target point, it adapts its velocity vector to stay in the middle between its two neighbors. This strategy can be viewed as arising from δ-GTM when $\delta \to 0$ with a speed limit of 1.

We will show that with all of the considered strategies, the robots converge towards or even reach a stable configuration with all robots being positioned on the line between the two base stations. This position a robot converges to or reaches will be called its *end position*.

2.4 Quality Measures

We want to measure the quality of the algorithms in terms of the number of rounds as well as the traveled distance.

When using the discrete time model and (variants of) GO-TO-THE-MIDDLE, the robots only converge to their end positions, and the number of rounds is unbounded. Instead, we will measure the number of rounds until all robots are in distance 1 of their end positions. For all strategies in the continuous time model, the number of rounds is not defined. We thus only analyze the maximum movement distance for these strategies, using this model to show how far the movement distance can be reduced when letting the number of rounds go to infinity.

Regarding the movement distance, we upper bound the maximum of the total distances traveled by the robots, the maximum taken over all robots. For the continuous strategies, the maximum traveled distance is a fixed value for a fixed start configuration, since the robots reach their end positions. When using δ-GTM or 1-GTM, the robots only converge to their end positions. But this means that the distance traveled by the robots also converges to a fixed value, and therefore we can upper bound it. In order to have comparable measures, we will lower bound the maximum traveled distance until all robots are in distance at most 1 from their end position. We will see that the upper and lower bounds for the traveled distance match asymptotically. That is, the distance traveled by the robots when they are already close to their end position can be neglected.

Chapter 3

The Robot Chain Problem: traveled distance

In this chapter we want to tackle the question how far the movement distance can be reduced, when the number of rounds is ignored. On the one hand, this shows how far we can hope to reduce the movement distance when considering both quality measures. On the other hand, this also makes sense if a robot type is used for which the movement distance is the major factor which determines the energy consumption of the robot, and which is able to continuously observe its environment. For the sake of analyzing the movement distance, we have introduced the Move-On-Bisector strategy in Section 2.3.1, which works in the continuous time model. We will first prove that it is valid in terms of connectedness: If neighbors are originally in distance at most 1, they will remain in distance at most 1 when performing the Move-On-Bisector-strategy. Then we show bounds on the maximum distance a robot can travel. Since the robots move with velocity 1 as long as they have not yet reached the line between their neighbors, this distance is equal to the time needed until all robots are on the straight line between the stations (the finishing time).

Unlike most strategies considered for similar problems, we use a continuous time model. Therefore, we are not given a classical round model, but rather all robots can perpetually and at the same time measure and adjust their movement paths, leading to curves as trajectories for the robots. Although this model fits to real applications [NPR$^+$03] and has also interesting and important theoretical aspects, surprisingly, to our knowledge, it has only once been considered theoretically for a formation problem [GWB04]. The authors give an algorithm which gathers robots in one point in finite time, but they do not give any further runtime bounds. One reason for not using a continuous time model might be that completely different techniques of analysis have to be applied than for usual discrete models. We are optimistic that the techniques for analysis which we develop in this thesis have the potential to be applied to other continuous formation problems. In Chapter 8 we show that they can be directly applied to the gathering problem.

This chapter is divided into three major parts: in Section 3.1 we show that the MOVE-ON-BISECTOR-strategy maintains a valid chain, such that the robots stay in viewing range of their neighbors. Then we analyze the time and equally the maximum traveled distance needed until all robots are positioned on the line between the two stations: First we show an upper bound for the maximum traveled distance of $\mathcal{O}(l)$. This bound is tight for configurations in which some robots are far away from their final destination, that is, the height is only by a constant factor smaller than the length of the chain (Section 3.2). Clearly, a global algorithm also needs long to optimize those chains. Moreover, since $l \leq n$, an upper bound of $\mathcal{O}(n)$ follows. Remember that h is a lower bound for the time needed by an optimal global algorithm. Since there are configurations with $h \in \Omega(n)$, this bound shows that MOVE-ON-BISECTOR is asymptotically optimal for worst-case configurations. In Section 3.3 we proceed to the main result. We show that configurations that are solved fast by an optimal global algorithm are also handled fast by MOVE-ON-BISECTOR. In particular, we show an upper bound of $\mathcal{O}((h+d)\log l)$. This bound shows that MOVE-ON-BISECTOR is $\mathcal{O}(\log n)$ competitive compared to an optimal global algorithm, if d is sufficiently small.

3.1 Validity of the Move-On-Bisector strategy

Let us first consider two robots v_i and v_j with $j > i$ at a time when neither v_i nor v_j have reached the line between their neighbors, but every robot v_k with $i < k < j$ has. That is, the robots v_k form a straight line between v_i and v_j. We will show that the distance between v_i and v_j decreases with non-negative speed. Given that all robots v_k between v_i and v_j maintain the ratio between the distances to their corresponding neighbors, this implies that the distance between *any* two neighboring robots is monotonically decreasing, and thus the chain stays connected and the MOVE-ON-BISECTOR-strategy is valid. We start by considering the case that both, v_i and v_j, are *mobile* (not stations).

Lemma 3.1. *Given two* robots *v_i and v_j as described above at an arbitrary time t_0, their distance decreases with speed* $\cos\frac{\alpha_i(t_0)}{2} + \cos\frac{\alpha_j(t_0)}{2} \geq 0$.

Proof. We define $D : \mathbb{R}_{\geq 0} \to \mathbb{R}^2, t \mapsto v_j - v_i$ and $\mathsf{d} : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}, t \mapsto ||D(t)||$. That is, $D(t)$ is the vector from v_i to v_j and $\mathsf{d}(t)$ the distance between v_i and v_j at time t. We want to show that $\mathsf{d}'(t_0) = -\left(\cos\frac{\alpha_i(t_0)}{2} + \cos\frac{\alpha_j(t_0)}{2}\right)$ for an arbitrary but

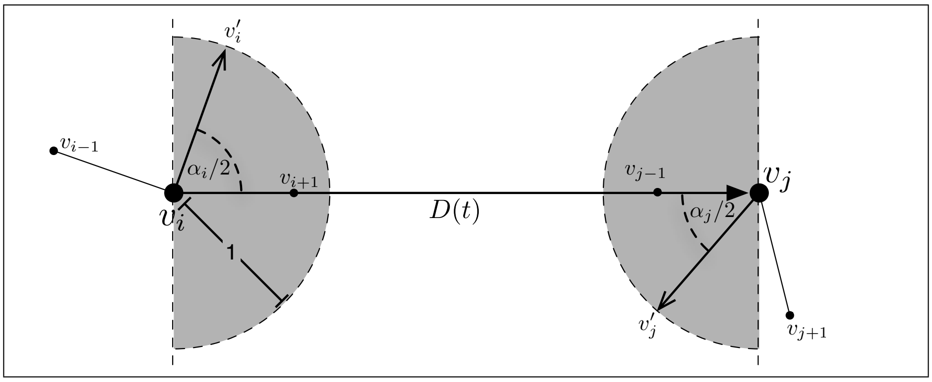


Figure 3.1: Illustration of v_i's and v_j's velocity vectors v_i' and v_j'.

fixed point of time t_0. We will refer to the x- and y-component of $D(t) \in \mathbb{R}^2$ in the following by $D_x(t)$ and $D_y(t)$ respectively.

By translating and rotating the coordinate system, we can w.l.o.g. assume $v_i(t_0) = (0,0)$ and $v_j(t_0) = (\mathsf{d}(t_0), 0)$. Due to the definition of the MOVE-ON-BISECTOR strategy, the velocity vectors of v_i and v_j at time t_0 are given by:

$$v_i'(t_0) = \left(+\cos\tfrac{\alpha_i(t_0)}{2}, \pm\sin\tfrac{\alpha_i(t_0)}{2}\right)$$

$$v_j'(t_0) = \left(-\cos\tfrac{\alpha_j(t_0)}{2}, \pm\sin\tfrac{\alpha_j(t_0)}{2}\right)$$

See Fig. 3.1 for an illustration.

Basic analysis now gives us the following equation for the first derivation of d at a time $t \in \mathbb{R}_{\geq 0}$[1]:

$$\mathsf{d}'(t) = \begin{pmatrix} \frac{D_x(t)}{\mathsf{d}(t)} & \frac{D_y(t)}{\mathsf{d}(t)} \end{pmatrix} \cdot \begin{pmatrix} D_x'(t) \\ D_y'(t) \end{pmatrix}$$

Using that we have $D_y(t_0) = 0$ and $D_x(t_0) = \mathsf{d}(t_0)$ we finally get

$$\begin{aligned} \mathsf{d}'(t_0) = D_x'(t_0) = &\ (v_j - v_i)'(t_0) = v_j'(t_0) - v_i'(t_0) \\ = &\ -\left(\cos\tfrac{\alpha_i(t_0)}{2} + \cos\tfrac{\alpha_j(t_0)}{2}\right). \end{aligned}$$

Therefore, the distance between v_i and v_j changes at time t with speed $\cos(\frac{\alpha_i(t_0)}{2}) + \cos(\frac{\alpha_j(t_0)}{2})$. Furthermore, since we have $\alpha_i(t) \in [0, \pi]$ for any $t \in \mathbb{R}_{\geq 0}$ and $i \in \{1, \ldots, n\}$, this speed is indeed positive and the distance *decreases*. ■

A similar result holds if either v_i or v_j is a station. Since this can be proven completely analogously to Lemma 3.1, we will omit the proof and merely state the corresponding result.

[1] Remember that we assume $\mathsf{d}(t) \neq 0$ (see the description of the MOVE-ON-BISECTOR-strategy in Section 2.3.1).

Lemma 3.2. *Consider two robots v_i and v_j at an arbitrary time t_0, one of them being a station and the other a robot not yet having reached the line between its neighbors. Then their distance decreases with speed* $\cos \frac{\alpha_j(t_0)}{2} \geq 0$.

Now, we have the preliminaries to state the validity of the MOVE-ON-BISECTOR strategy.

Theorem 3.3. *The* MOVE-ON-BISECTOR *strategy is valid. That is, if the robot chain is connected at time t and all robots perform the* MOVE-ON-BISECTOR *strategy, the robot chain remains connected for any time $t' \geq t$.*

Proof. As described above, the statement follows immediately from Lemma 3.1 and 3.2 together with the fact that any robot that has already reached the line between its neighbors will move such that it maintains the ratio between the distances to its two neighbors. ■

3.2 The $\mathcal{O}(l)$ upper bound

We continue by analyzing how long it will take for all robots to reach the straight line between the two stations. We will derive a time bound of $\mathcal{O}(l)$, l denoting the length of the robots' initial configuration. Because $h \leq l/2$ and $l = \mathcal{O}(n)$ (the distance of neighboring robots is bounded by 1), this immediately implies a linear bound $\mathcal{O}(n)$ on the time until the optimal configuration is reached. Since there are start configurations with a height of $\Omega(n)$ and h is a lower bound even for the time needed by an optimal global algorithm, the MOVE-ON-BISECTOR-strategy is asymptotically optimal for worst case start configurations. The next section will show a tighter bound for configurations, where the height is relatively small compared to the length of the configuration.

In the following, we will show that either the length l or the height h of the robot chain decreases with constant speed. Since both are furthermore monotonically decreasing and bounded from below, this implies that the optimum configuration will be reached in time $\mathcal{O}(h + l) = \mathcal{O}(l)$. We begin with the monotonicity of the height.

Lemma 3.4. *The height of the robot chain is monotonically decreasing and bounded from below by 0.*

Proof. The lower bound is trivial, it follows directly from the definition of the robot chain's height. For the monotonicity, fix a time $t \in \mathbb{R}_{\geq 0}$ and consider the height $\mathfrak{h}(t)$ of the configuration at time t. Let B denote the line segment connecting both stations and note that all robots are contained in the convex set $H := \{x \in \mathbb{R}^2 \mid dist(x, B) \leq \mathfrak{h}(t)\}$ of points having a distance of at most $\mathfrak{h}(t)$ to B (see Fig. 2.2). Let us consider an arbitrary robot v_k and its neighbors v_{k-1} and v_{k+1}. Since H is convex and all three robots lie in H, so does the bisector along which v_k moves. That is, v_k cannot leave H. Since this argument applies to any robot, none of the robots can increase their distance to B beyond $\mathfrak{h}(t)$. This implies the monotonicity of the robot chain's height. ■

Lemma 3.5. *The length of the robot chain decreases with speed* $2\sum_{i=1}^{n} \cos \frac{\alpha_i(t)}{2}$ *and is bounded from below by* d.

Proof. Since both stations do not move, the length obviously cannot fall below their distance d. Using the function $\mathfrak{l} : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}, t \mapsto \mathfrak{l}(t)$ to refer to the chain's length at time t, it remains to show that $\mathfrak{l}'(t) = -2\sum_{i=1}^{n} \cos \frac{\alpha_i(t)}{2}$.

Fix a time $t \in \mathbb{R}_{\geq 0}$ and consider the robots $v_{i_1}, v_{i_2}, \ldots, v_{i_r}$ (for an $r \in \mathbb{N}$ and $i_s < i_{s+1} \forall s = 1, \ldots, r-1$) that have not yet reached the line between their neighbors. We make two observations:

- For any robot v_j of the remaining robots, it holds that $\alpha_j(t) = \pi$ and therefore $\cos \frac{\alpha_j(t)}{2} = 0$.
- Any of the remaining robots either lies on the line between some v_{i_s} and $v_{i_{s+1}}$ or on the line between one of the stations and v_{i_1} or v_{i_r}. That is, setting $\mathfrak{l}_0(t) := ||v_0(t) - v_{i_1}(t)||$, $\mathfrak{l}_k(t) := ||v_{i_k}(t) - v_{i_{k+1}}(t)||$ $(k = 1, \ldots, r-1)$ and $\mathfrak{l}_r(t) := ||v_{i_r}(t) - v_{n+1}(t)||$, the length $\mathfrak{l}(t)$ of the chain is given by:

$$\mathfrak{l}(t) = \sum_{k=0}^{r} \mathfrak{l}_k(t)$$

Now, Lemma 3.1 and Lemma 3.2 give us the derivations of these $\mathfrak{l}_k$, and therefore we have:

$$\begin{aligned}
\mathfrak{l}'(t) &= \mathfrak{l}_0'(t) + \sum_{k=1}^{r-1} \mathfrak{l}_k'(t) + \mathfrak{l}_r'(t) \\
&= -\cos\frac{\alpha_{i_1}(t)}{2} + \sum_{k=1}^{r-1}\left(-\cos\frac{\alpha_{i_k}(t)}{2} - \cos\frac{\alpha_{i_{k+1}}(t)}{2}\right) - \cos\frac{\alpha_{i_r}(t)}{2} \\
&= -2\sum_{k=1}^{r}\cos\frac{\alpha_{i_k}(t)}{2} = -2\sum_{i=1}^{n}\cos\frac{\alpha_i(t)}{2}.
\end{aligned}$$

■

Now we can prove an upper bound for the traveled distance in dependency of h and l, implying also a worst case upper bound.

Theorem 3.6. *When the* MOVE-ON-BISECTOR *strategy in the continuous model is performed, the maximum distance traveled by a robot is upper bounded by* $\sqrt{2}h + \frac{1}{\sqrt{2}}l$.

Proof. We will prove that in time $\sqrt{2}h + \frac{1}{\sqrt{2}}l$ all robots have reached their corresponding end positions. Given that the robots move with a maximum velocity of 1, this proves the theorem. To do so, we show that at any time, either the height function $\mathfrak{h} : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ or the length function $\mathfrak{l} : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ are strictly decreasing *by a constant factor*. Together with Lemma 3.4 and Lemma 3.5 (the monotonicity and non-negativity of $\mathfrak{l}$ and $\mathfrak{h}$) this proves the theorem.

So, let us consider an arbitrary time $t \in \mathbb{R}_{\geq 0}$. We distinguish two cases:

Case 1: $\exists i \in \{1, \ldots, n\} : \alpha_i(t) \leq \pi/2$

In this case, Lemma 3.5 states that:

$$\mathfrak{l}'(t) = -2 \sum_{k=1}^{n} \cos \frac{\alpha_k(t)}{2} \leq -2 \cos \frac{\alpha_i(t)}{2} \leq -2 \cos \frac{\pi}{4} = -\sqrt{2}$$

That is, the length of the robot chain decreases with a *constant* speed of at least $\sqrt{2}$. Since the length cannot drop below d, after time at most $\frac{1}{\sqrt{2}}l$ with an angle $\alpha_i(t) \leq \pi/2$ the length of the chain is d and all robots must have stopped moving.

Case 2: $\forall i \in \{1, \ldots, n\} : \alpha_i(t) > \pi/2$

Using the terms from the proof of Lemma 3.4, consider a robot v_k with distance $\mathfrak{h}(t)$ to the line segment B connecting both stations. Align the coordinate system such that the line L through $v_k(t)$ having distance $\mathfrak{h}(t)$ to B corresponds to the x axis and $v_k(t)$ to the origin. Fig. 3.2 illustrates the situation.

We know that both neighbors of v_i must lie on the same side of L as B, w.l.o.g. let it be the lower side. Furthermore, because we have $\alpha_k(t) > \pi/2$, one neighbor must lie to the lower left and the other to the lower right of v_k. This implies that v_k's velocity vector is directed downwards, forming an angle of less than $\pi/4$ with the y-axis. Therefore, v_k moves with a speed of more than $\cos \frac{\pi}{4}$ downwards.

Since this holds for any extremal robot, we get $h'(t) < -\cos \frac{\pi}{4} = -\frac{1}{\sqrt{2}}$. That is, the height of the robot chain decreases with a constant speed of at least $\frac{1}{\sqrt{2}}$ and after time $\sqrt{2}h$ with $\alpha_i(t) > \pi/2$ for all robots i the height has decreased to 0.

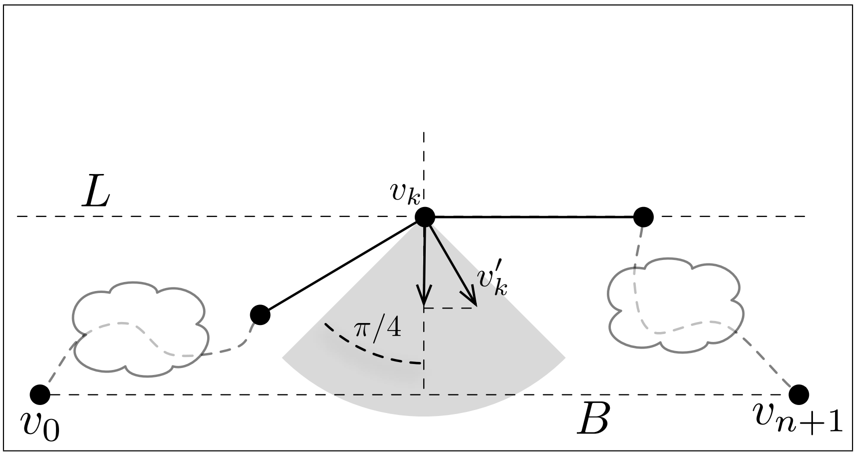


Figure 3.2: If all angles α_i are larger than $\pi/2$, then the velocity vector of a "highest" robot v_k lies within the gray area. It therefore moves downwards with a speed of at least $\cos \pi/4$.

■

Since $h \in \mathcal{O}(l)$, Theorem 3.6 gives an upper bound of $\mathcal{O}(l)$ for arbitrary start configurations. This result directly shows that the Move-On-Bisector-strategy is asymptotically optimal for worst-case instances (Corollary 3.7), the measure which is usually used in the literature. Still, this bound can be arbitrarily worse than an optimal algorithm on specific instances. We will investigate these instances in the next section.

Corollary 3.7. *When the* Move-On-Bisector-*strategy in the continuous model is performed, the maximum distance traveled by a robot is* $\Theta(n)$ *for a worst-case start configuration.*

Proof. Obviously it holds that $h \leq \frac{l}{2} \leq \frac{n+1}{2}$. For the lower bound, we can use a start configuration in which the stations share the position $(0,0)$ and $v_i(0) = v_{n+1-i}(0) = (0,i)$. Thus, the robot in the middle of the chain is in distance $\approx \frac{n}{2}$ of its end position and Move-On-Bisector (as well as any global algorithm) needs at least this time until all robots have reached the line between the stations. ■

3.3 The $\mathcal{O}((h+d)\log l)$ upper bound

Assume we are given a configuration whose height is - relative to the length of the robot chain - very small. In this case, the upper bound of $\mathcal{O}(l)$ for our strategy can be arbitrarily larger than the time needed by an optimal strategy, which can

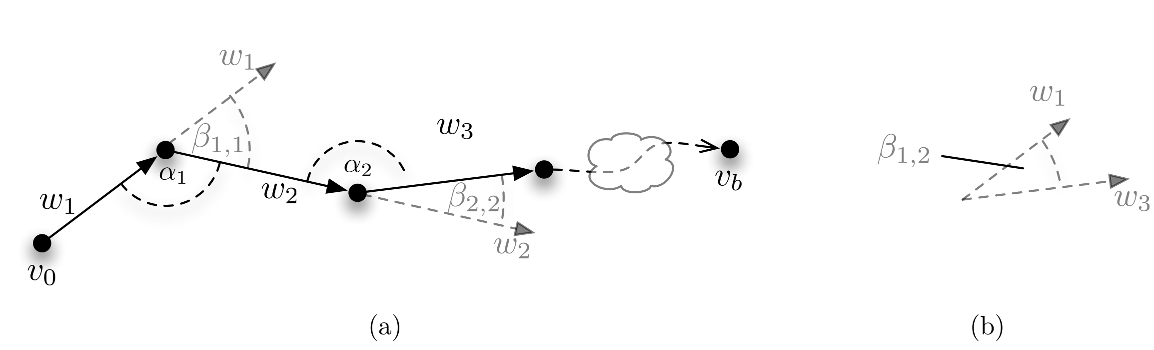


Figure 3.3: Note that the angles $\beta_{i,j}$ are signed, e.g.: $\beta_{1,1} > 0$, $\beta_{2,2} < 0$, $\beta_{1,2} = \beta_{1,2} + \beta_{2,3} > 0$.

be as small as h. But intuitively, given a long chain with a small height, the chain must be quite winding, yielding many relatively small angles α_i. The result is that the chain length does not only decrease at one robot, as we can only guarantee for arbitrary configurations, but there are many robots which reduce the length of the chain (Lemma 3.5).

For the proof of this upper bound, we will divide the chain into parts of length $\Theta(h + d)$ and show that each part must contain some curves. In particular, in each part, the sum of the angles $\alpha_i(t)$ must be by a constant smaller than in a straight line (Lemma 3.8). Lemma 3.9 transfers this result for each part to the sum of the angles of the whole chain. Having that the sum of the angles in the whole chain cannot be arbitrarily large, Lemma 3.10 yields the speed by which the length of the chain decreases. Since the number of parts is dependent on the length of the chain, the speed is also dependent on it. Theorem 3.11 finally gives the upper bound of $\mathcal{O}((d + h) \log l)$.

Lemma 3.8. *Let $\mathfrak{B}$ denote an arbitrary rectangular box containing the robots v_{a-1}, v_a, $v_{a+1}, \ldots, v_b$ (for $a, b \in \{1, \ldots, n+1\}$, $a < b$) at a given time $t \in \mathbb{R}_{>0}$ and let S be the diagonal length of the box. Then we have:*

$$\sum_{k=a}^{b} ||w_k(t)|| \geq \sqrt{2} \cdot S \Rightarrow \sum_{k=a}^{b-1} \alpha_k(t) \leq (b-a)\pi - \frac{\pi}{3}$$

Proof. For the sake of clarity, we will omit the time parameter t in the following. That is we write α_k, v_k and w_k instead of $\alpha_k(t)$, $v_k(t)$ and $w_k(t)$. Furthermore, we assume w.l.o.g. $a = 1$. Thus, we have to show $\sum_{k=1}^{b} ||w_k|| \geq \sqrt{2} \cdot S \Rightarrow \sum_{k=1}^{b-1} \alpha_k \leq (b-1)\pi - \frac{\pi}{3}$

Consider the function $\angle : \mathbb{R}^2 \times \mathbb{R}^2 \to]-\pi, \pi]$ that maps two vectors (w_i, w_j) to the signed angle of absolute value $\leq \pi$ formed by them (it is not important which direction is used as positive angle, as long as it is equal for all pairs of vectors (w_i, w_j)). Note that we have $\alpha_k = \pi - |\angle(w_k, w_{k+1})|$ for all $k = 1, \dots, b-1$. Let us define $\beta_{i,j} := \sum_{k=i}^{j} \angle(w_k, w_{k+1})$ and observe that $\angle(w_i, w_j) \equiv \beta_{i,j} \mod]-\pi, \pi]$. See Fig. 3.3 for an illustration.

Let us now assume $\sum_{k=1}^{b} ||w_k(t)|| \geq \sqrt{2} \cdot S$ and consider the following two cases:

Case 1: $\exists i, j, 1 \leq i < j \leq b : |\beta_{i,j}| \geq \frac{\pi}{3}$

Intuitively, if the angle between two vectors in the chain is large, the sum of the inner angles α_k of the robots in between cannot be arbitrarily large. More formally,

$$\begin{aligned}
\sum_{k=1}^{b-1} \alpha_k &\leq (i-1)\pi + \sum_{k=i}^{j} \alpha_k + (b-1-j)\pi \\
&= (b+i-j-2)\pi + \sum_{k=i}^{j} (\pi - |\angle(w_k, w_{k+1})|) = (b-1)\pi - \sum_{k=i}^{j} |\angle(w_k, w_{k+1})| \\
&\leq (b-1)\pi - \left| \sum_{k=i}^{j} \angle(w_k, w_{k+1}) \right| = (b-1)\pi - |\beta_{i,j}| \leq (b-1)\pi - \frac{\pi}{3}
\end{aligned}$$

Thus, the lemma holds in this case.

Case 2: $\forall i, j, 1 \leq i < j \leq b : |\beta_{i,j}| < \frac{\pi}{3}$

We will show that this case cannot occur by showing that the vector connecting v_0 and v_b, which is equal to $\sum_{k=1}^{b} w_k$, would have to be longer than S, which is a contradiction to v_0 and v_b both lying in $\mathfrak{B}$.

We have $\angle(w_i, w_j) = \beta_{i,j}$ and $|\beta_{i,j}| < \frac{\pi}{3}$ for all $1 \leq i < j \leq b$. In the following, we will use that the squared length of a vector is equal to its scalar product with itself. Therefore:

$$\begin{aligned}
\left|\left| \sum_{k=1}^{b} w_k \right|\right|^2 &= \left(\sum_{k=1}^{b} w_k \right) \cdot \left(\sum_{k=1}^{b} w_k \right) = \sum_{1 \leq i,j \leq b} w_i \cdot w_j = \sum_{1 \leq i,j \leq b} ||w_i|| \cdot ||w_j|| \cdot \cos(\beta_{i,j}) \\
&> \sum_{1 \leq i,j \leq b} ||w_i|| \cdot ||w_j|| \cdot \cos(\frac{\pi}{3}) = \cos(\frac{\pi}{3}) \sum_{1 \leq i,j \leq b} ||w_i|| \cdot ||w_j|| \\
&= \frac{1}{2} \left(\sum_{k=1}^{b} ||w_k|| \right)^2 \geq \frac{1}{2} \cdot (\sqrt{2}S)^2 = S^2
\end{aligned}$$

This implies $|| \sum_{k=1}^{b} w_k || > S$, leading to the desired contradiction.

■

Dividing the chain in parts of length at least $\sqrt{2}$ times the diagonal of the height box, Lemma 3.8 shows that each of the parts must contain some "small" angles. The robots at these angles therefore shorten the length of the chain. The following lemma shows that using the technique of dividing the chain into parts yields an upper bound on the sum of the angles α_i of the chain.

Lemma 3.9. *Let S denote the diagonal length of the robots' height-box at a given time t. Then we have:*

$$\sum_{k=1}^{n} \alpha_k(t) \leq n\pi - \frac{\pi}{3} \left\lfloor \frac{\mathfrak{l}(t)}{2\sqrt{2}S} \right\rfloor$$

Proof. As in the proof for Lemma 3.8, we will omit the time parameter t in the following.

First note that we have $||w_k|| \leq S$, because all robots lie inside the height-box. This allows us to recursively define indices $1 = a_0 < a_1 < \ldots < a_m \leq n+1$ by demanding $a_i \in \mathbb{N}$ to be minimal with $\sum_{k=a_{i-1}}^{a_i} ||w_k|| \in [\sqrt{2}S, (\sqrt{2}+1)S[$. That is, we divide the chain at time t in m parts, where $v_{a_{i-1}}$ and v_{a_i} bound part i. v_{a_i} is the first robot in the chain such that the length of part i is at least $\sqrt{2}S$. Furthermore, since $||w_{a_i}|| \leq S$, the length of part i is at most $\sqrt{2}S + S \leq 2\sqrt{2}S$, which implies $m \geq \left\lfloor \frac{\mathfrak{l}}{2\sqrt{2}S} \right\rfloor$. Since we have $\sum_{k=a_{i-1}}^{a_i} ||w_k|| \geq \sqrt{2}S$ for all $i = 1, \ldots, m$, by Lemma 3.8 we get $\sum_{k=a_{i-1}}^{a_i-1} \alpha_k \leq (a_i - a_{i-1})\pi - \frac{\pi}{3}$. We now compute:

$$\begin{aligned}
\sum_{k=1}^{a_m-1} \alpha_k &= \sum_{i=1}^{m} \sum_{k=a_{i-1}}^{a_i-1} \alpha_k \leq \sum_{i=1}^{m} \left((a_i - a_{i-1})\pi - \frac{\pi}{3}\right) \\
&= \pi \sum_{i=1}^{m} (a_i - a_{i-1}) - \frac{\pi}{3} m = (a_m - a_0)\pi - \frac{\pi}{3} m \\
&= (a_m - 1)\pi - \frac{\pi}{3} m
\end{aligned}$$

This implies $\sum_{k=1}^{n} \alpha_k \leq n\pi - \frac{\pi}{3} m \leq n\pi - \frac{\pi}{3} \left\lfloor \frac{\mathfrak{l}}{2\sqrt{2}S} \right\rfloor$, as the lemma states. ■

Using that the sum of the angles α_i is bounded, we can now give a lower bound for the speed by which the chain length decreases, which is linear in the current number of parts and therefore the length of the chain. Instead of the current number of parts, which cannot be determined exactly only knowing the length of the chain, we use a lower bound for the number of parts.

Lemma 3.10. *The length of the robot chain decreases at least with speed $\frac{2}{3} \left\lfloor \frac{\mathfrak{l}(t)}{2\sqrt{2}S} \right\rfloor$.*

Proof. Fix a time $t \in \mathbb{R}_{\geq 0}$. By Lemma 3.5, the chain length decreases with a speed of $2\sum_{k=1}^{n} \cos \frac{\alpha_k(t)}{2}$. Using that $\cos(x)$ is lower bounded by $1 - \frac{2}{\pi}x$ for all $x \in [0, \pi/2]$ and by Lemma 3.9 we get:

$$\begin{aligned}\mathfrak{l}'(t) &= -2\sum_{k=1}^{n} \cos \frac{\alpha_k(t)}{2} \leq -2\sum_{k=1}^{n}\left(1 - \frac{\alpha_k(t)}{\pi}\right) = -2n + \frac{2}{\pi}\sum_{k=1}^{n} \alpha_k(t) \\ &\leq -2n + \frac{2}{\pi}\left(n\pi - \frac{\pi}{3}\left\lfloor \frac{\mathfrak{l}(t)}{2\sqrt{2}S} \right\rfloor\right) = -\frac{2}{3}\left\lfloor \frac{\mathfrak{l}(t)}{2\sqrt{2}S} \right\rfloor.\end{aligned}$$

■

Now we can finally state our main result.

Theorem 3.11. *When the* MOVE-ON-BISECTOR *strategy in the continuous model is performed, the maximum distance traveled by a robot is $\mathcal{O}((h+d)\log(l))$, where h is the height and l the length of the robot chain in the start configuration.*

Proof. Set $m^* := \left\lfloor \frac{l}{2\sqrt{2}S} \right\rfloor$ and let us define m^* time-phases $\mathfrak{p}_i := [t_{i-1}, t_i]$ for $i = 1 \ldots, m^*$ by setting $t_0 := 0$ and t_i for $i > 0$ to the time when we have $\mathfrak{l}(t_i) = (m^* - i + 1) \cdot 2\sqrt{2}S$. That is, during one phase $\mathfrak{p}_i$, the chain length is reduced by exactly $2\sqrt{2}S$ and thus in phase i, the chain must consist of at least m^* parts as defined in Lemma 3.9. Note that these t_i are well-defined, because by Lemma 3.10, in phase $\mathfrak{p}_i$ the chain length decreases with a speed of at least $\frac{2}{3}\left\lfloor \frac{\mathfrak{l}(t_i)}{2\sqrt{2}S} \right\rfloor = \frac{2}{3} \cdot (m^* - i + 1)$ (which is a constant for fixed i). Furthermore, Lemma 3.10 gives us an upper bound on the length of each single phase $\mathfrak{p}_i$:

$$t_i - t_{i-1} \leq \frac{\mathfrak{l}(t_{i-1}) - \mathfrak{l}(t_i)}{\frac{2}{3}(m^* - i + 1)} \leq \frac{2\sqrt{2}S}{\frac{2}{3}(m^* - i + 1)}.$$

This allows us to give an upper bound to the time when the last phase ends:

$$\begin{aligned}t_{m^*} &= \sum_{i=1}^{m^*}(t_i - t_{i-1}) \leq 3\sqrt{2}S\sum_{i=1}^{m^*} \frac{1}{m^* - i + 1} \\ &= 3\sqrt{2}S\sum_{i=1}^{m^*} i^{-1} < 3\sqrt{2}S \cdot (\ln m^* + 1).\end{aligned}$$

Now consider the situation after time $t \geq t_{m^*}$. We have $\mathfrak{l}(t_{m^*}) = (m^* - m^* + 1)2\sqrt{2}S = 2\sqrt{2}S$. By Theorem 3.6, from now on it takes time at most $\sqrt{2}\mathfrak{h}(t_{m^*}) + \frac{1}{\sqrt{2}}\mathfrak{l}(t_{m^*}) \leq \sqrt{2}h + 2S$ for the robots to reach the optimal configuration. Together with the bound on t_{m^*} and with $S = \mathcal{O}(h+d)$, this yields a maximum time (and therefore traveled distance) of

$$
\begin{aligned}
&3\sqrt{2} \cdot S \cdot (\ln m^* + 1) + \sqrt{2}h + 2S \\
\leq &3\sqrt{2} \cdot S \cdot \left(\ln\left(\frac{l}{2\sqrt{2}S}\right) + 1\right) + \sqrt{2}h + 2S \\
= &3\sqrt{2} \cdot S \cdot (\ln l - \ln(2\sqrt{2}S) + 1) + \sqrt{2}h + 2S \\
= &\mathcal{O}(S \cdot \ln l) + \sqrt{2}h + 2S = \mathcal{O}((h + d)\ln l)
\end{aligned}
$$

until the optimal configuration is reached. ■

Corollary 3.12. *Using* MOVE-ON-BISECTOR, *the maximum distance traveled by the robots is* $\mathcal{O}(min\{n, (OPT + d)\log n\})$.

A consequence of this result is that, for $d \in \mathcal{O}(h)$, our local algorithm is by at most a logarithmic factor slower than an optimal global algorithm.

C H A P T E R 4

The Robot Chain Problem: both quality measures

In Chapter 3 we have seen that the distance traveled when using a local algorithm can be relatively close to the distance which needs to be traveled when using an optimal global algorithm, when the robots may observe their environment continuously and the number of rounds and Look-Operations is infinite. Now we want to combine both quality measures. The first obvious question is what distance the robots travel when using the original GO-TO-THE-MIDDLE strategy 1-GTM. We will see that it is $\Theta(n^2)$ for worst case instances and therefore a lot worse than that of MOVE-ON-BISECTOR. Therefore, we reduce the step size of GO-TO-THE-MIDDLE to $\delta \in (0, 1]$ and analyze the movement distance as well as the number of rounds of the resulting δ-GTM strategy. We will see that this technique helps indeed to improve the quality of the algorithm. When choosing $\delta \in \Theta(1/n)$, the movement distance is reduced to $\mathcal{O}(n)$ and therefore asymptotically optimal for worst case instances, while the needed number of rounds is still $\mathcal{O}(n^2 \log n)$ as for 1-GTM. Thus, no trade-off between the two quality measures is required. For an overview of the results confer to Table 4.1.

Table 4.1: Results overview

	1-bounded GTM	δ-bounded GTM	continuous GTM
number of time steps	$\Theta(n^2 \log n)$ [Kut07, KM11]	$\Omega(n^2 + \frac{n}{\delta})$ Thm. 4.12 $\mathcal{O}(n^2 \log n + \frac{n}{\delta})$ Thm. 4.12	—
maximum distance	$\Theta(n^2)$ Cor. 4.23	$\Theta(\delta n^2 + n)$ Thm. 4.22	$\Theta(n)$ Thm. 4.26

Section 4.1 is devoted to the δ-GTM strategy, with 1-GTM being a special case. It gives lower and upper bounds for the worst-case number of rounds as well as for the maximum distance traveled by a robot. In Section 4.2 we analyze Continuous-GTM

in the continuous time model. We will see that similar to MOVE-ON-BISECTOR, Continuous-GTM achieves a maximum traveled distance of $\mathcal{O}(n)$ for worst case start configurations.

4.1 The δ-bounded Go-To-The-Middle strategy

In this section, we consider the 1-bounded GO-TO-THE-MIDDLE strategy and the δ-bounded GO-TO-THE-MIDDLE strategy. With 1-GTM being a special case of the δ-GTM for $\delta = 1$, we will limit the analysis to δ-GTM, whose results can be easily adapted.
In δ-GTM, robots only move up to a distance of δ per round. Thus, they are not always able to reach their target point within one round.

A major observation for the sake of analysis is the fact that for δ-GTM, we can divide the movement of the robots into two phases. In the first phase at least one of the robots is not able to reach its target point. In the second phase, the target point of every robot lies within a δ distance of its current position. Thus, every robot reaches its target point, while the target point moves a distance of at most δ. Therefore every robot is still able to reach it in the next round. Thus we stay in the second phase once we have reached it. Since every robot reaches its target point, the second phase is indistinguishable from the 1-bounded GO-TO-THE-MIDDLE strategy and therefore the validity follows from [Kut07]. But we still need to show that the first phase also keeps the chain connected. The proof is a straightforward adaption of the proof of validity of 1-GTM in [Kut07], which covers the special case for $\delta = 1$.

Theorem 4.1. *δ-GTM is a valid strategy. That is, if the chain is connected in round 0, then, when applying δ-GTM, the chain is connected after every round.*

Proof. We will show that the chain is always connected by induction on the rounds t. It is connected in round 0 by assumption. So now assume that the chain is connected at the end of round t: $|v_i(t) - v_{i+1}(t)| \leq 1$ for all $i = 0, ..., n$. We will show that the movement of the robots during round $t+1$ does not break the connectivity and thus, $|v_i(t+1) - v_{i+1}(t+1)| \leq 1$ for all $i = 0, ..., n$.

Fix two arbitrary neighboring robots v_i and v_{i+1} and call e the line segment between $v_i(t)$ and $v_{i+1}(t)$. Translate and rotate the coordinate system such that e is completely on the x-axis and that the origin is in the middle of e (see Fig. 4.1). Let $a := |e|/2$.

We draw a circle with radius $1/2$ around the origin of the coordinate system. We will show that after applying δ-GTM, both $v_i(t+1)$ and $v_{i+1}(t+1)$ are within this circle, and so the distance between them is at most 1.

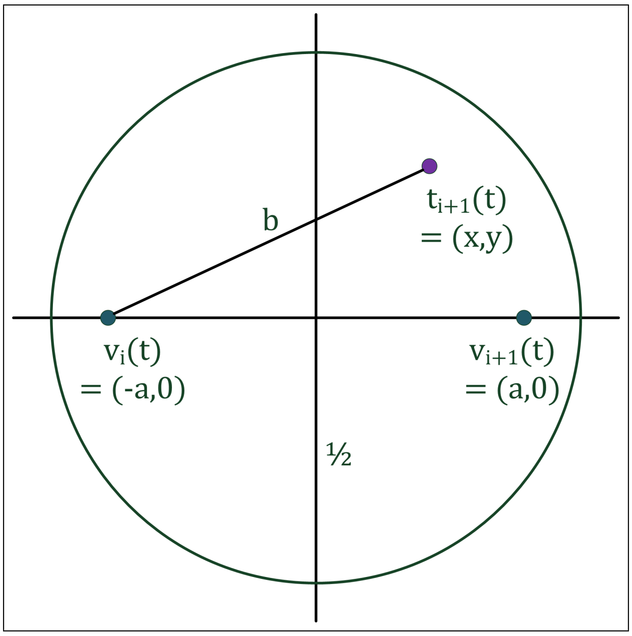


Figure 4.1: Illustration to the proof of Theorem 4.1

Define $t_{i+1}(t) = (x, y)$ to be the coordinates of v_{i+1}'s target point in round t and $b := |v_i(t) - t_{i+1}(t)|$ to be the distance between v_i in round t and v_{i+1}'s target point. From the Pythagorean theorem we have $b = \sqrt{(x+a)^2 + y^2}$.

Since $t_{i+1}(t)$ is the target point of v_{i+1} in round t, it is in the middle between $v_i(t)$ and $v_{i+2}(t)$. It follows that $v_{i+2}(t) = (2x + a; 2y)$.

Now assume for the sake of contradiction that $t_{i+1}(t) = (x, y)$ lies outside the circle. Then $x^2 + y^2 > (1/2)^2$. We now compute the distance $|v_{i+1}(t) - v_{i+2}(t)|$. We have

$$\begin{aligned} |v_{i+1}(t) - v_{i+2}(t)| &= \sqrt{(2x+a-a)^2 + (2y-0)^2} = \sqrt{4 \cdot (x^2+y^2)} \\ &> \sqrt{4 \cdot \frac{1}{4}} = 1. \end{aligned}$$

This contradicts the assumption that the chain is connected in round t. Thus, v_{i+1}'s target point is positioned in the circle. Since the circle is convex, it follows that all points on the line between v_{i+1} and its target point are also inside the circle, and so is the point $v_{i+1}(t+1)$ to which v_{i+1} moves in round $t+1$.

The same reasoning can be applied to determine the position of $v_i(t+1)$. Thus, we have proven that both $v_i(t+1)$ and $v_{i+1}(t+1)$ are in distance at most 1 from each other. ■

We will now start by analyzing the number of rounds and then investigate the maximum traveled distance.

4.1.1 The worst-case number of rounds

In this subsection we analyze the number of rounds, before analyzing the maximum distance traveled by a robot in the next subsection. The number of rounds can be divided into the number of rounds of the two phases. Thus, we first analyze the first phase. We start with a lower bound, before presenting a matching upper bound. The lower bound only holds for $\delta \leq \frac{1}{n}$, while the upper bound only holds for $\delta \geq \frac{1}{n}$. We will see that only in this case the number of rounds is dominated by the respective phase.

First, we define two types of configurations.

Definition 4.2. *A configuration $\mathcal{C}$ at time t is in a* triangular-like shape, *if the following conditions hold. The two stations are positioned in distance d on the x-axis, such that $v_0(t) = (0,0)$, $v_{n+1}(t) = (d,0)$. The positions of the n mobile robots, n uneven, are as follows. Concerning the x-coordinate, each robot is positioned in the middle between its neighbors. Regarding the y-coordinate, the middle robot $v_{\frac{n+1}{2}}$ has the largest distance to the target line. In particular, $y_{\frac{n+1}{2}}(t) \geq y_{\frac{n-1}{2}}(t) \geq y_{\frac{n-3}{2}}(t) \geq \ldots \geq y_1(t) \geq 0$. Moreover, the configuration is symmetric: $y_1(t) = y_n(t), y_2(t) = y_{n-1}(t), \ldots, y_{\frac{n+1}{2}-1}(t) = y_{\frac{n+1}{2}+1}(t)$.*

See Fig. 4.3 and Fig. 4.4 for examples of configurations in a triangular-like shape. Note that d can be chosen arbitrarily small.

Definition 4.3. *A configuration in a* triangular shape *is a configuration in a triangular-like shape, where the robots $v_1, \ldots, v_{\frac{n+1}{2}-1}$ and $v_{\frac{n+1}{2}+1}, \ldots, v_n$ are positioned in the middle between their neighbors, such that they form a triangle.*

Figure 4.3 is an example for a configuration in a triangular shape.

The first phase: Lower bound

We will need the following lemma in order to show a lower bound for the number of rounds for worst-case configurations until all robots are in distance at most δ from their target point and therefore until we reach the second phase.

Lemma 4.4. *Consider a configuration in a triangular-like shape with all robots being in the second phase, and a robot v_k in k hops from the middle robot $v_{\frac{n+1}{2}}$. Then the y-distance between v_k and $v_{\frac{n+1}{2}}$ is at most $k^2 \cdot \delta$.*

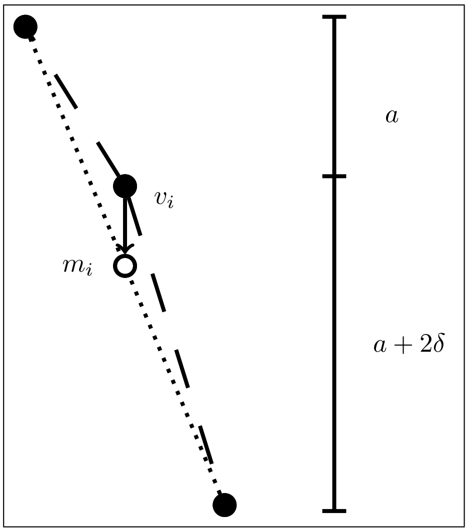


Figure 4.2: The y-distances to the neighbors can differ by at most 2δ

Proof. This proof is divided into two parts. First we will show that a robot with k hops to $v_{\frac{n+1}{2}}$ has a distance of at most $(2 \cdot k - 1) \cdot \delta$ to its neighbors. In the second part we will use this to prove the proposition.

Proof by induction: Base case $i = 1$. Because the robots are in a triangular-like shaped configuration, $v_{\frac{n+1}{2}}$ has the same y-distance to both its neighbors. Therefore it has the same y-distance to its target point, which is in the middle between its neighbors. Since moreover $v_{\frac{n+1}{2}}$ may be at most in distance δ from its target point, the y-distance of $v_{\frac{n+1}{2}}$ and its neighbors can be at most δ.
Induction step: Let v_i be a robot in i hops from $v_{\frac{n+1}{2}}$. Its neighbor v_{i-1}, which is in $i-1$ hops from $v_{\frac{n+1}{2}}$, is in y-distance at most $a := (2i-3)\,\delta$ from $v_{\frac{n+1}{2}}$. Now consider Fig. 4.2. Since v_i's target point must be within a δ distance of v_i, robot v_i can be in y-distance most $a + 2\delta = (2i-1)\,\delta$ from $v_{\frac{n+1}{2}}$.
Now we can simply add the values to obtain the maximum distance of v_k to $v_{\frac{n+1}{2}}$.

$$\begin{aligned}\sum_{i=1}^{k}(2i-1)\cdot\delta &= \delta\cdot\sum_{i=0}^{k-1}(2i+1) = 2\delta\sum_{i=0}^{k-1} i + \delta\cdot\sum_{i=0}^{k-1} 1 = 2\delta\cdot\frac{k\cdot(k-1)}{2} + \delta k \\ &= \delta k^2 - \delta k + \delta k = \delta k^2\end{aligned}$$

■

Lemma 4.5. *There is a start configuration for which the number of rounds in the first phase of δ-GTM is $\Omega(\frac{n}{\delta})$ for $\delta \leq \frac{1}{n}$.*

Proof. Consider a a triangular shaped start configuration with robot $v_{\frac{n+1}{2}}$ at the top (see Fig. 4.3). We will show that after $\frac{1}{16}\cdot\frac{n}{\delta}$ rounds at least one robot still has

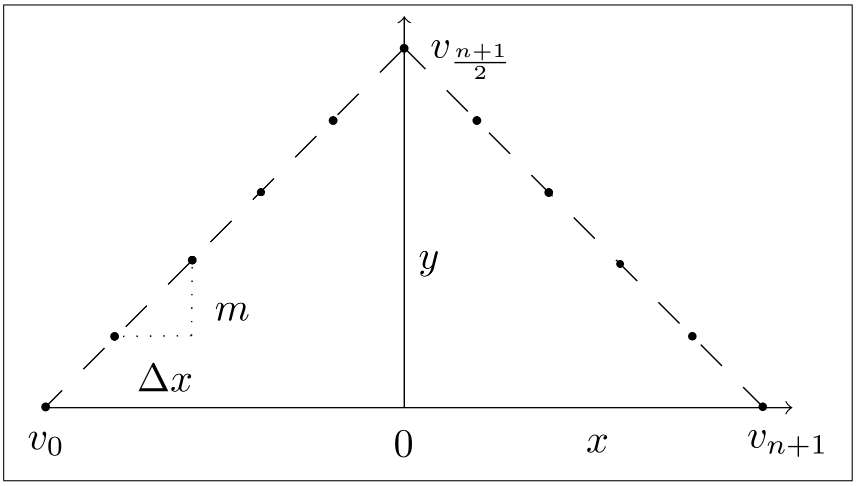


Figure 4.3: Start configuration in a triangular shape

a distance which is greater than δ to its target point. We will do so by showing that the distance of $v_{\frac{n+1}{2}}$ to its end position (called its *height* from now on) after $\frac{1}{16} \cdot \frac{n}{\delta}$ rounds is greater than its height if the target point of every robot is within a δ distance from the robot itself. First we will calculate its height in the start configuration: $m \cdot \frac{n+1}{2} = \frac{7}{8} \cdot \frac{n+1}{2}$ where $m := \frac{7}{8}$ is the constant y-distance between two neighbors. Note that we can choose $m \leq 1$ arbitrarily, since we can choose the x-distance between two neighbors arbitrarily small. After $\frac{1}{16} \cdot \frac{n}{\delta}$ rounds with step size δ its remaining height is at least $\frac{7}{8} \cdot \frac{n+1}{2} - \frac{1}{16} \cdot n \geq \frac{7}{16} \cdot n - \frac{1}{16} \cdot n = \frac{3}{8}n$.

Now consider the maximum height $v_{\frac{n+1}{2}}$ can have if every robot is within a δ distance of its target point. Note that the robots are still in a triangular-like shaped configuration. Due to Lemma 4.4, a robot with k hops to $v_{\frac{n+1}{2}}$ can have a maximum y-distance from $v_{\frac{n+1}{2}}$ of $\delta \cdot k^2$. Since the stations are in $\frac{n+1}{2}$ hops from $v_{\frac{n+1}{2}}$, $v_{\frac{n+1}{2}}$ can have a height of at most $\left(\frac{n+1}{2}\right)^2 \cdot \delta = \frac{n^2}{4}\delta + \frac{n}{2}\delta + \frac{1}{4}\delta$.

But since $\frac{n^2}{4}\delta + \frac{n}{2}\delta + \frac{1}{4}\delta \leq \frac{n}{4} + \frac{1}{2} + \frac{1}{4n} < \frac{1}{4}n + \frac{1}{8}n = \frac{3}{8}n$ with $\delta \leq \frac{1}{n}$ and $n \geq 5$, there must be at least one robot which has a distance greater than δ from its target point. ■

The first phase: Upper bound

After having shown a lower bound of $\Omega\left(\frac{n}{d}\right)$ for $\delta \leq \frac{1}{n}$, we will now prove an upper bound of $\mathcal{O}\left(\frac{n}{d}\right)$ and thus $\Theta\left(\frac{n}{d}\right)$ for the first phase. We start with a technical lemma. Let $y_i := y_i(0)$ be the y-coordinate of robot v_i in the start configuration, and $\Delta_i = \frac{y_{i-1}+y_{i+1}}{2} - y_i$ be the positive or negative y-distance of robot v_i to its target point in the start configuration.

Lemma 4.6. $-1 \leq \sum_{i=j}^{k} \Delta_i \leq 1$ *holds for for arbitrary* $1 \leq j \leq k \leq n$.

Proof. The distance is defined as

$$\Delta_i = \frac{y_{i-1} + y_{i+1}}{2} - y_i$$

Thus for the sum we obtain:

$$\begin{aligned}
\left|\sum_{i=j}^{k} \Delta_i\right| &= \left|\sum_{i=j}^{k}\left(\frac{y_{i-1}+y_{i+1}}{2} - y_i\right)\right| \\
&= \left|\frac{1}{2}\sum_{i=j}^{k} y_{i-1} + \frac{1}{2}\sum_{i=j}^{k} y_{i+1} - \sum_{i=j}^{k} y_i\right| = \left|\frac{1}{2}\sum_{i=j-1}^{k-1} y_i + \frac{1}{2}\sum_{i=j+1}^{k+1} y_i - \sum_{i=j}^{k} y_i\right| \\
&= \left|\sum_{i=j+1}^{k-1} y_i - \sum_{i=j}^{k} y_i + \frac{1}{2}\left(y_{j-1} + y_j + y_k + y_{k+1}\right)\right| \\
&= \frac{1}{2}\left|(y_{j-1} - y_j + y_{k+1} - y_k)\right| \leq \frac{1}{2}\left|(y_{j-1} - y_j)\right| + \frac{1}{2}\left|(y_{k+1} - y_k)\right| \\
&\leq 1
\end{aligned}$$

■

Lemma 4.7. *The number of rounds in the first phase of* δ*-GTM is* $\mathcal{O}(\frac{n}{\delta})$.

Proof. Assume for the sake of contradiction that there is a robot v_i which moves $9 \cdot \frac{n}{\delta}$ rounds with step size δ without reaching its target point. We will show that this would require the robots which have a station as one of their neighbors to move more than they are able to. We first show that the maximum distance traveled by robot v_1 and v_n respectively is limited. This can be done because the target points of those robots move at most with step size $\frac{\delta}{2}$. Because of that v_1 and v_n will only move with step size $\frac{\delta}{2}$ after having reached their target point for the first time. Moreover, before having reached their target point for the first time, the distance between v_1 and v_n and their target point decreases by at least $\frac{\delta}{2}$ in each round. Thus, after $\frac{2}{\delta}$ rounds and a traveled distance of 2, they have definitely reached their target points. This results in an upper bound for the traveled distance of v_1 and v_n after t rounds of $2 + t \cdot \frac{\delta}{2}$.

Now we will show that the distance v_1 or v_n need to travel, if v_i does not reach its target point after $9 \cdot \frac{n}{\delta}$ rounds, is larger than that (for $n \geq 8$). If v_i does not reach its target point, it travels a distance of $9n$. W.l.o.g. assume that v_n is closer to v_i than v_1 (or in equal distance). Now let s_j be the distance traveled by the robot v_{i+j}, which is in j hops from v_i, in the first $9 \cdot \frac{n}{\delta}$ rounds. That is, as v_i travels a total distance of $s_0 = 9n$. Let there be k robots with a larger index than v_i, meaning that v_n travels a distance of s_k.

Proposition 4.8. *Let ∇_l, Δ_l be the x-distance respectively the y-distance of robot v_{i+l} to its target point in the start configuration. Then*

$$s_k \geq 9n - 2 \cdot \textstyle\sum_{l=0}^{k-1}(k-l) \cdot \Delta_l - 2 \cdot \sum_{l=0}^{k-1}(k-l) \cdot \nabla_l.$$

The proof of this proposition follows after the proof of this lemma. As v_n is closer to v_i than v_1, we know that $k \leq \frac{n}{2}$. Plugging this in, we get

$$\begin{aligned} s_k &\geq 9n - 2 \cdot \sum_{l=0}^{\frac{n}{2}-1}(\frac{n}{2} - l) \cdot \Delta_l - 2 \cdot \sum_{l=0}^{\frac{n}{2}-1}(\frac{n}{2} - l) \cdot \nabla_l \\ &= 9n - n\sum_{l=0}^{\frac{n}{2}-1} \Delta_l + 2\sum_{l=0}^{\frac{n}{2}-1} l \cdot \Delta_l - n\sum_{l=0}^{\frac{n}{2}-1} \nabla_l + 2\sum_{l=0}^{\frac{n}{2}-1} l \cdot \nabla_l \end{aligned}$$

We now use one further structural property of the robot chain, saying that $-1 \leq \sum_{i=j}^{k} \Delta_i \leq 1$ and $-1 \leq \sum_{i=j}^{k} \nabla_i \leq 1$ for arbitrary $1 \leq j \leq k \leq n$ (Lemma 4.6):

$$\begin{aligned} s_k &\geq 9n - n\sum_{l=0}^{\frac{n}{2}-1} \Delta_l + 2\sum_{l=0}^{\frac{n}{2}-1} l \cdot \Delta_l - n\sum_{l=0}^{\frac{n}{2}-1} \nabla_l + 2\sum_{l=0}^{\frac{n}{2}-1} l \cdot \nabla_l \\ &\geq 9n - n + 2\sum_{l=1}^{\frac{n}{2}-1}\sum_{i=l}^{\frac{n}{2}-1} \Delta_i - n + 2\sum_{l=1}^{\frac{n}{2}-1}\sum_{i=l}^{\frac{n}{2}-1} \nabla_i \\ &\geq 7n + 2\sum_{l=1}^{\frac{n}{2}-1} -1 + 2\sum_{l=1}^{\frac{n}{2}-1} -1 \geq 5n. \end{aligned}$$

We are now able to put the pieces together. Either v_1 or v_n must move a distance of $5n$ to allow v_i to move a distance of $9n$. At the same time, the distance v_1 and v_n can move is upper bounded by $2 + t \cdot \frac{\delta}{2} = 2 + 9\frac{n}{\delta} \cdot \frac{\delta}{2} = 2 + \frac{9}{2}n$. This is a contradiction for $n \geq 8$. Therefore no robot can move $9 \cdot \frac{n}{\delta}$ rounds without reaching its target point and this means that the number of rounds in the first phase is bounded by $\mathcal{O}\left(\frac{n}{\delta}\right)$. ■

Proof of Proposition 4.8. The idea of the proof is that if v_i moves a distance of $9n$ without reaching its target point, the target point must move a distance of at least $9n - \Delta_0 - \nabla_0$. But the movement of v_i's target point results from the movement of v_i's neighbors, such that their movement can be lower bounded. This kind of argumentation propagates through the complete chain.

We start the proof at robot v_{i+k} and determine its movement distance depending on the distance its left neighbor moves. In particular, we show by induction for $1 \leq j \leq k-1$ that

$$s_k \geq (j+1)s_{k-j} - j \cdot s_{k-(j+1)} - 2 \cdot \sum_{l=1}^{j} l \cdot \Delta_{k-l} - 2 \cdot \sum_{l=1}^{j} l \cdot \nabla_{k-l}. \tag{4.1}$$

Base case $j = 1$: We need to show that $s_k \geq 2 \cdot s_{k-1} - s_{k-2} - 2 \cdot \Delta_{k-1} - 2 \cdot \nabla_{k-1}$. For this, consider the movement of v_{i+k}'s left neighbor. It can move at most the distance its target point moves, which is upper bounded by $\frac{1}{2}s_k + \frac{1}{2}s_{k-2}$, plus its distance to its target point in the start configuration. Thus, $s_{k-1} \leq \frac{1}{2}s_k + \frac{1}{2}s_{k-2} + \Delta_{k-1} + \nabla_{k-1}$. This gives that $s_k \geq 2s_{k-1} - s_{k-2} - 2\Delta_{k-1} - 2\nabla_{k-1}$.

For the inductive step, we know by the induction hypothesis that

$$s_k \geq (j+1)s_{k-j} - j \cdot s_{k-(j+1)} - 2 \cdot \sum_{l=1}^{j} l \cdot \Delta_{k-l} - 2 \cdot \sum_{l=1}^{j} l \cdot \nabla_{k-l}. \tag{4.2}$$

Moreover, we can again use the movement of $v_{i+(k-j-1)}$ and its target point to reformulate s_{k-j}. Like in the base case, $s_{k-j-1} \leq \frac{1}{2}s_{k-j} + \frac{1}{2}s_{k-j-2} + \Delta_{k-j-1} + \nabla_{k-j-1}$, giving that $s_{k-j} \geq 2s_{k-j-1} - s_{k-j-2} - 2\Delta_{k-j-1} - 2\nabla_{k-j-1}$. Plugging this in into (4.2), we get

$$\begin{aligned}
s_k &\geq (j+1)\left(2s_{k-j-1} - s_{k-j-2} - 2\Delta_{k-j-1} - 2\nabla_{k-j-1}\right) - j \cdot s_{k-(j+1)} \\
&\quad -2 \cdot \sum_{l=1}^{j} l \cdot \Delta_{k-l} - 2 \cdot \sum_{l=1}^{j} l \cdot \nabla_{k-l} \\
&= (j+2)s_{k-(j+1)} - (j+1) \cdot s_{k-(j+2)} - 2 \cdot \sum_{l=1}^{j+1} l \cdot \Delta_{k-l} - 2 \cdot \sum_{l=1}^{j+1} l \cdot \nabla_{k-l},
\end{aligned}$$

yielding the inductive step.

Now we can express the movement of s_k using s_0 and s_1: Set $j = k-1$. Then, according to (4.1), $s_k \geq k \cdot s_1 - (k-1) \cdot s_0 - 2 \cdot \sum_{l=1}^{k-1} l \cdot \Delta_{k-l} - 2 \cdot \sum_{l=1}^{k-1} l \cdot \nabla_{k-l}$.

Finally, we show a lower bound for the movement of v_{i+1}. Consider the movement of v_i: If v_i moves a distance of $s_0 = 9n$ without reaching its target point, its target point has to move a distance of at least the distance which v_i moves minus $\Delta_0 + \nabla_0$, an upper bound on initial distance between v_i and its target point. The movement of v_i's target point results from a movement of its neighbors: $9n - \Delta_0 - \nabla_0 \leq \frac{1}{2}s_{-1} + \frac{1}{2}s_1$. s_{-1} denotes the distance v_{i-1} moves and can be upper bounded by $9n$ ($9\frac{n}{\delta}$ rounds with a distance of at most δ. Thus, $s_1 \geq 9n - 2\Delta_0 - 2\nabla_0$.

Plugging this in, we get

$$\begin{aligned}
s_k &\geq k \cdot (9n - 2\Delta_0 - 2\nabla_0) - (k-1) \cdot 9n - 2 \cdot \sum_{l=1}^{k-1} l \cdot \Delta_{k-l} - 2 \cdot \sum_{l=1}^{k-1} l \cdot \nabla_{k-l} \\
&= -2\Delta_0 k - 2\nabla_0 k + 9n - 2 \cdot \sum_{l=1}^{k-1} l \cdot \Delta_{k-l} - 2 \cdot \sum_{l=1}^{k-1} l \cdot \nabla_{k-l} \\
&= 9n - 2 \cdot \sum_{l=1}^{k} l \cdot \Delta_{k-l} - 2 \cdot \sum_{l=1}^{k} l \cdot \nabla_{k-l} \\
&= 9n - 2 \cdot \sum_{l=0}^{k-1} (k-l) \cdot \Delta_l - 2 \cdot \sum_{l=0}^{k-1} (k-l) \cdot \nabla_l,
\end{aligned}$$

yielding the proposition. ■

The second phase

We have now analyzed the number of rounds of the first phase. In order to state the overall number of rounds, we now analyze the second phase as well. We will see that the second phase can only take longer for $\delta \geq 1/n$. We can therefore simplify the analysis of the second phase by assuming $\delta \geq 1/n$. For this, as a progress measure we use the sum of the y-distances $y_i(t)$ from the robots to the target line, define it to be $Y(t)$. We construct a configuration for which $Y(0)$ is $\Omega(n^2)$, and we use that it decreases by at most 1 in each round [Kut07]. It follows that $\Omega(n^2)$ rounds are required until the sum is at most n, which is a necessary condition for the robots being in distance at most 1 from their end position.

We start with showing that $Y(t)$ decreases by at most 1 per round.

Lemma 4.9 ([Kut07]). *If all robots are on the same side of the target line,*

$$Y(t) - Y(t+1) = \frac{|y_1(t)| + |y_n(t)|}{2}.$$

Proof. Since all robots are on the same side of the target line, we have $|y_i(t)| = y_i(t)$. Then we get

$$\begin{aligned} Y(t+1) &= \sum_{i=1}^{n} y_i(t+1) = \sum_{i=1}^{n} \frac{1}{2}(y_{i-1}(t) + y_{i+1}(t)) \\ &= \sum_{i=0}^{n-1} \frac{1}{2} y_i(t) + \sum_{i=2}^{n+1} \frac{1}{2} y_i(t) \\ &= \sum_{i=1}^{n} y_i(t) - \frac{1}{2} y_1(t) - \frac{1}{2} y_n(t) = Y(t) - \frac{1}{2} y_1(t) - \frac{1}{2} y_n(t) \end{aligned}$$

The lemma follows. ■

Corollary 4.10. *If all robots are on the same side of the target line, $Y(t)$ decreases by at most 1 per round.*

Now we can show the lower bound for the number of rounds.

Lemma 4.11. *There is a start configuration such that when δ-GTM is used and $\delta \geq \frac{1}{n}$, the number of rounds in the second phase until each robot is in distance at most 1 from its end position is $\Omega(n^2)$.*

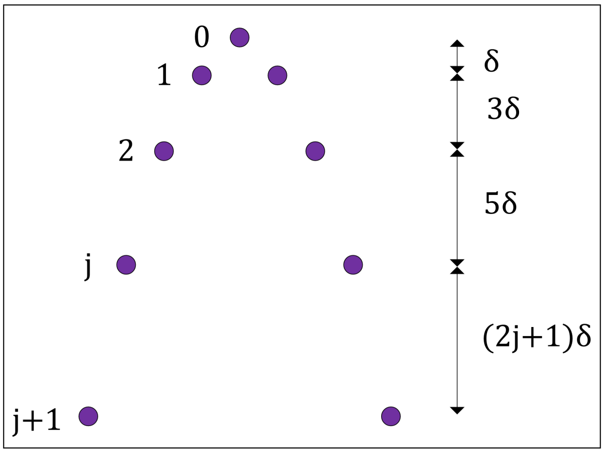


Figure 4.4: The upper part of the start configuration for Lemma 4.11

Proof. Consider a triangular-like shaped start configuration consisting of two parts. The middle robot and those robots which can be reached from the middle robot in $\frac{1}{2\delta} - 1$ hops form the upper part, whereas the remaining robots form the lower part. The y-coordinates of the robots in the upper part are as follows: The robot in j hops from the middle robot has a y-distance of $(2j-1)\delta$ to the robot in $j-1$ hops from the middle robot and a y-distance of $(2j+1)\delta$ to the robot in $j+1$ hops from the middle robot. See Fig. 4.4 for a visualization of the upper part. For each robot i of the lower part holds $y_i = y_{i-1} + 1$ (v_i is left of the middle robot) or $y_i = y_{i+1} + 1$ (v_i is right of the middle robot) and thus these robots form a line. See Fig. 4.5 for an illustration of the complete configuration. The distance d between the two stations must be 0 here for the robots in the lower part to be connected. For better illustration, it is drawn to be greater than 0 in the figure.

The robots in the upper part are in distance at most δ from their target point. Except for the robots $v_{\frac{n-1}{2}-\frac{1}{2\delta}}$ and $v_{\frac{n-1}{2}+\frac{1}{2\delta}}$, which are positioned at the border of the upper and lower part, the robots in the lower part are already positioned at their target point. Robot $v_{\frac{n-1}{2}-\frac{1}{2\delta}}$ is in distance 1 of its left neighbor and in distance $(2(\frac{1}{2\delta}-1)+1)\delta = 1-\delta$ of its right neighbor. Therefore, the distance between robot $v_{\frac{n-1}{2}-\frac{1}{2\delta}}$ and its target point is $\frac{\delta}{2}$. The same holds for robot $v_{\frac{n-1}{2}+\frac{1}{2\delta}}$. Accordingly, the robots are already in the second phase of δ-GTM.

As $y_i(t)$ is positive for all robots, $Y(t) := \sum_{i=1}^{n} y_i(t)$. We show that $Y(0) \in \Omega(n^2)$ and we use that $Y(t)$ decreases by at most 1 in every round. We will follow that $\Omega(n^2)$ rounds are required until each robot is in distance at most 1 from its end position.

To compute $Y(0)$ for our start configuration, we first introduce the notion of a *row*. Let i be the number of hops from a robot v_k to the middle robot, where v_k is

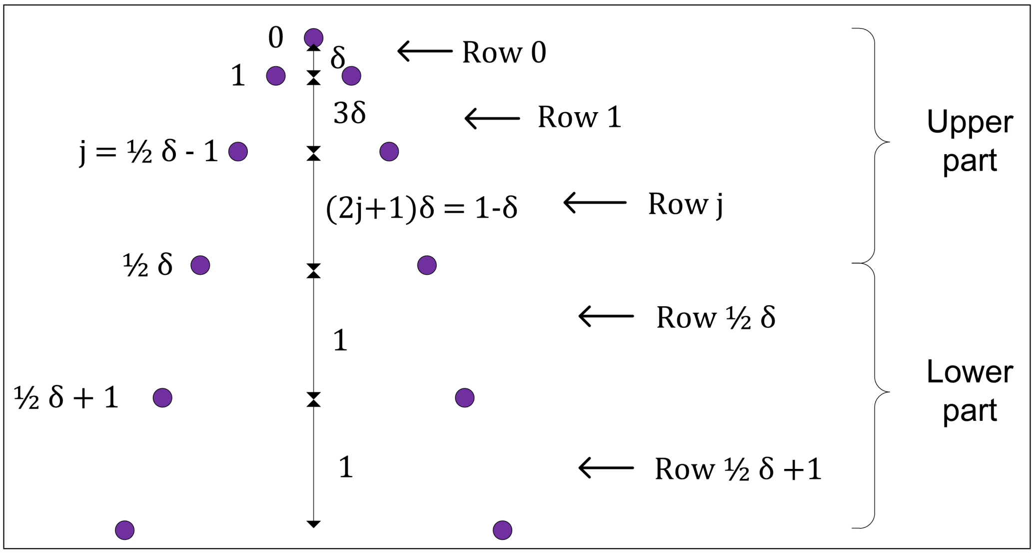


Figure 4.5: Complete start configuration for Lemma 4.11

positioned left of the middle robot. $row_i := (y_k - y_{k-1}) \cdot (2i+1)$ is then the y-distance between v_k and v_{k-1} times the number of robots which have a y-coordinate of at least y_k. Like this, we can rewrite $Y(0) = \sum_{i=0}^{\frac{n-1}{2}} row_i$. This sum can now be splitted in the upper and the lower part: $\sum_{i=0}^{\frac{n-1}{2}} row_i = \sum_{i=0}^{\frac{1}{2\delta}-1} row_i + \sum_{i=\frac{1}{2\delta}}^{\frac{n-1}{2}} row_i$. We now compute these sums independently, starting with the upper part. We have $2i+1$ robots in row i, and the height of the row is $(2i+1) \cdot \delta$.

$$\begin{aligned}
\sum_{i=0}^{\frac{1}{2\delta}-1} row_i &= \sum_{i=0}^{\frac{1}{2\delta}-1} (2i+1)^2\delta = \delta \sum_{i=0}^{\frac{1}{2\delta}-1} (4i^2+4i+1) = \delta(\frac{1}{2\delta} + 4 \sum_{i=0}^{\frac{1}{2\delta}-1} i^2 + 4 \sum_{i=0}^{\frac{1}{2\delta}-1} i) \\
&= \frac{1}{2} + 4\delta \frac{(\frac{1}{2\delta}-1)\frac{1}{2\delta}(\frac{1}{\delta}-1)}{6} + 4\delta \frac{(\frac{1}{2\delta}-1)\frac{1}{2\delta}}{2} \\
&= \frac{1}{2} + \frac{2}{3}\delta(\frac{1}{4\delta^2} - \frac{1}{2\delta})(\frac{1}{\delta}-1) + \frac{1}{2\delta} - 1 \\
&= \frac{1}{2} + (\frac{1}{6\delta} - \frac{1}{3})(\frac{1}{\delta}-1) + \frac{1}{2\delta} - 1 \\
&= \frac{1}{2} + \frac{1}{6\delta^2} - \frac{1}{6\delta} - \frac{1}{3\delta} + \frac{1}{3} + \frac{1}{2\delta} - 1 = \frac{1}{6\delta^2} - \frac{1}{6}
\end{aligned}$$

The sum for the lower part can be computed as follows. Here, the first inequality follows from having $2i+1$ robots in row i, where each row has height 1.

$$\begin{aligned}\sum_{i=\frac{1}{2\delta}}^{\frac{n-1}{2}} row_i &= \sum_{i=\frac{1}{2\delta}}^{\frac{n-1}{2}} (2i+1) \geq \sum_{i=\frac{1}{2\delta}}^{\frac{n-1}{2}} 2i = 2\cdot\sum_{i=0}^{\frac{n-1}{2}} i - 2\cdot\sum_{i=0}^{\frac{1}{2\delta}-1} i \\ &= 2\frac{\frac{n-1}{2}\frac{n+1}{2}}{2} - 2\cdot\frac{(\frac{1}{2\delta}-1)\frac{1}{2\delta}}{2} = \frac{n^2-1}{4} - \frac{1}{4\delta^2} + \frac{1}{2\delta} \\ &\geq \frac{n^2}{4} - \frac{1}{4} - \frac{1}{4\delta^2}\end{aligned}$$

Now we can combine the upper and lower part to get a lower bound for $Y(0)$, using that $\delta \geq 1/n$.

$$Y(0) \geq \frac{1}{6\delta^2} - \frac{1}{6} + \frac{n^2}{4} - \frac{1}{4} - \frac{1}{4\delta^2} = \frac{n^2}{4} - \frac{5}{12} - \frac{1}{12\delta^2} \geq \frac{n^2}{4} - \frac{n^2}{12} - \frac{5}{12} = \frac{n^2}{6} - \frac{5}{12}.$$

Now we have a lower bound for the start value of our progress measure $Y(t)$. Due to Corollary 4.10 we know that $Y(t)$ decreases by at most 1 per round, and therefore after t rounds we have $Y(t) \geq Y(0) - t \geq \frac{n^2}{6} - \frac{5}{12} - t$.

Moreover, it follows that there exists a robot v_i with $y_i(t) \geq \frac{1}{n}\cdot(\frac{n^2}{6} - \frac{5}{12} - t)$. Thus, for $y_i(t)$ to be less than or equal to 1, it must hold

$$1 \geq \frac{1}{n}\cdot(\frac{n^2}{6} - \frac{5}{12} - t) \Leftrightarrow t \geq \frac{n^2}{6} - n - \frac{5}{12},$$

and thus $t \in \Omega(n^2)$. ■

Conclusion

We can now conclude the results about the worst case number of rounds used by δ-GTM.

Theorem 4.12. *For a worst-case start configuration, the number of rounds for δ-GTM until each robot is in distance at most 1 from its end position is $\Omega(n^2 + \frac{n}{\delta})$ and $\mathcal{O}(n^2 \log n + \frac{n}{\delta})$.*

Proof. According to Lemma 4.5, the number of rounds in the first phase when starting in a worst-case configuration is $\Omega(\frac{n}{\delta})$, if $\delta \leq \frac{1}{n}$. Moreover, Lemma 4.11 gives a lower bound for the second phase of $\Omega(n^2)$, if $\delta \geq \frac{1}{n}$. Combined, we get a lower bound of $\max\{\Omega(n^2), \Omega(\frac{n}{\delta})\} = \Omega(n^2 + \frac{n}{\delta})$, because $\frac{n}{\delta} \geq n^2$ if and only if $\delta \leq \frac{1}{n}$.

The upper bound also consists of the first phase, which takes $\mathcal{O}(\frac{n}{\delta})$ rounds according to Lemma 4.7, and the second phase. Since in the second phase δ-GTM does not differ from 1-GTM, the upper bound of $\mathcal{O}(n^2 \log n)$ from [Kut07] holds here as well. Thus we get an upper bound of $\mathcal{O}(n^2 \log n + \frac{n}{\delta})$. ■

Interpreting this result, we can see that the second phase takes longer if $\delta \in o(\frac{1}{n\log n})$ and the first phase if $\delta \in \omega(\frac{1}{n})$. For $\delta \in \Omega(\frac{1}{n\log n})$ and $\delta \in O(\frac{1}{n})$, it is still unclear which phase takes longer. For the triangle configuration in Fig. 4.3, even an optimal global strategy needs $\Theta(\frac{n}{\delta})$ rounds and thus for $\delta \in \mathcal{O}(\frac{1}{n\log n})$, δ-GTM is asymptotically optimal compared to an optimal global algorithm. To minimize the number of rounds, $\delta \in \Omega(\frac{1}{n\log n})$ should be chosen, resulting in a number of rounds of $\mathcal{O}(n^2 \log n)$.

We will now see that similar results hold for the maximum traveled distance.

4.1.2 Maximum distance traveled by a robot

In the last subsection we have analyzed the strategy with respect to the number of rounds. Now we analyze the the worst-case distance a robot has to travel. We start with an upper bound for the distance traveled in the second phase in an infinite number of rounds, before showing a lower bound for the maximum distance traveled in the second phase until all robots are at most in distance 1 from their end positions. We will see that these bounds match asymptotically.

Upper bound

Lemma 4.13. *In an infinite number of rounds in the second phase, a robot can move at most distance $\frac{1}{4}\delta n^2 + \frac{1}{2}\delta n + \frac{1}{4}\delta$ when using δ-GTM.*

Proof. Let $d_i(t)$ be the distance traveled by robot v_i in round t and let $d_i = \sum_{t=1}^{\infty} d_i(t)$. For all i, $1 \le i \le n$, $d_i(t) \le \frac{1}{2}d_{i-1}(t-1) + \frac{1}{2}d_{i+1}(t-1)$ and therefore

$$\begin{aligned}\sum_{t=2}^{\infty} d_i(t) &\le \frac{1}{2}\sum_{t=1}^{\infty} d_{i-1}(t) + \frac{1}{2}\sum_{t=1}^{\infty} d_{i+1}(t)\\ &= \frac{1}{2}d_{i-1} + \frac{1}{2}d_{i+1}.\end{aligned}$$

Since $d_i(1) \le \delta$, we get that

$$d_i \le \delta + \frac{1}{2}d_{i-1} + \frac{1}{2}d_{i+1} \tag{4.3}$$

Moreover, both stations do not move and thus $d_0 = d_{n+1} = 0$. Plugging this into (4.3), we get that $d_1 \le \delta + \frac{1}{2}d_2$. We can continue this for d_2, ..., d_n and get the following proposition for odd n:

Proposition 4.14. *1. For $1 \leq i < \frac{n}{2}$, $d_i \leq \delta i + \frac{i}{i+1} d_{i+1}$*

2. $d_{\lceil \frac{n}{2} \rceil} \leq \delta + d_{\lceil \frac{n}{2} \rceil - 1}$

3. For $\frac{n}{2} + 1 < i \leq n$, $d_i \leq \delta(n-i) + \frac{n-i}{n-i+1} d_{i-1}$.

Proof. Proof of case 1: Proof by induction.
Base case: $i = 1$. Because of (4.3) in the proof of Lemma 4.13, $d_1 \leq \delta + \frac{1}{2}d_0 + \frac{1}{2}d_2 = \delta + \frac{1}{2}d_2$.
For the inductive step, we know that

$$\begin{aligned} d_{i+1} &\leq \delta + \frac{1}{2}d_i + \frac{1}{2}d_{i+2} \\ &\leq \delta + \frac{i}{2}\delta + \frac{i}{2(i+1)}d_{i+1} + \frac{1}{2}d_{i+2} \\ \Leftrightarrow (1 - \frac{i}{2(i+1)})d_{i+1} &\leq \frac{i+2}{2}\delta + \frac{1}{2}d_{i+2} \\ \Leftrightarrow d_{i+1} &\leq (i+1)\delta + \frac{i+1}{i+2}d_{i+2}, \end{aligned}$$

where the first inequality follows from inequality (4.3) and the second from the induction hypothesis.

Case 2 deals with the middle robot: Because of the symmetry, the upper bound for the movement distance of its neighbors is equal. Case 3 is symmetric to case 1. Here, the second station is closer to the robots than the first one. So the proposition follows. ∎

It follows that the middle robot $v_{\lceil \frac{n}{2} \rceil}$ can move the furthest. With Proposition 4.14 we can upper bound $d_{\lceil \frac{n}{2} \rceil}$:

Proposition 4.15. *If n is odd, $d_{\lceil \frac{n}{2} \rceil} \leq \frac{\delta}{4}n^2 + \frac{\delta}{2}n + \frac{\delta}{4}$.*

Proof. Because of Proposition 4.14 we know that $d_{\lceil \frac{n}{2} \rceil} \leq \delta + d_{\lceil \frac{n}{2} \rceil - 1}$ and $d_{\lceil \frac{n}{2} \rceil - 1} \leq \delta(\lceil \frac{n}{2} \rceil - 1) + \frac{\lceil \frac{n}{2} \rceil - 1}{\lceil \frac{n}{2} \rceil} d_{\lceil \frac{n}{2} \rceil}$. Moreover, since n is odd, $\lceil \frac{n}{2} \rceil = \frac{n}{2} + \frac{1}{2}$.
This gives

$$\begin{aligned} d_{\lceil \frac{n}{2} \rceil} &\leq \delta + \delta(\lceil \frac{n}{2} \rceil - 1) + \frac{\lceil \frac{n}{2} \rceil - 1}{\lceil \frac{n}{2} \rceil} d_{\lceil \frac{n}{2} \rceil} \\ \Leftrightarrow \frac{1}{\lceil \frac{n}{2} \rceil} d_{\lceil \frac{n}{2} \rceil} &\leq \delta \lceil \frac{n}{2} \rceil \\ \Leftrightarrow d_{\lceil \frac{n}{2} \rceil} &\leq \delta \lceil \frac{n}{2} \rceil^2 = \delta(\frac{n}{2} + \frac{1}{2})^2 = \frac{\delta}{4}n^2 + \frac{\delta}{2}n + \frac{\delta}{4} \end{aligned}$$

∎

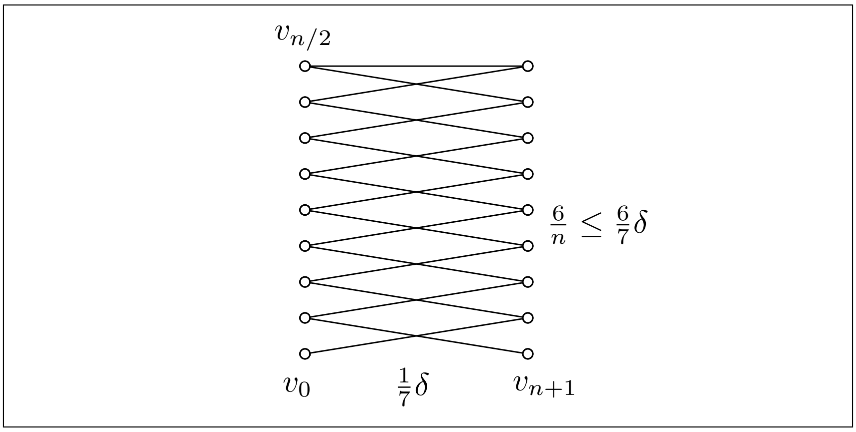


Figure 4.6: Example configuration of a tower

If n is even, we have two middle robots, which changes some details. Completely equivalently to Proposition 4.14 and Proposition 4.15, an upper bound for the movement distance of these robots can be computed, giving that $d_{\frac{n}{2}} \le \frac{\delta}{4}n^2 + \frac{\delta}{2}n$ for even n. Thus, no robot can move further than a distance of $\frac{\delta}{4}n^2 + \frac{\delta}{2}n + \frac{\delta}{4}$. ■

Lower bound

For the lower bound on the maximum traveled distance, consider the following start configuration, which we call a *tower* (see Fig. 4.6). The two stations are d apart. The n robots, where n is even, are placed alternatingly above the stations with an increase in height of $\frac{6}{n}$ for the first $n/2$ and a decrease of $\frac{6}{n}$ in height for the last $n/2$ robots, such that each robot has a y-distance of $\frac{6}{n}$ to one or both of its neighbors. For convenience's sake, we assume that v_0 is positioned at $(0,0)$ and v_{n+1} at $(d,0)$. Let $\delta \ge \frac{7}{n}$. Now we can define d as $\frac{\delta}{7}$ (which is at least $\frac{1}{n}$). Apparently, every robot can reach its target point and so we are in the second phase.

Now let $d_i(t)$ denote the distance traveled in x-direction from robot i in round t.

Lemma 4.16. *For the tower start configuration, $d_i(t) = \frac{1}{2}d_{i-1}(t-1) + \frac{1}{2}d_{i+1}(t-1)$ for all rounds t and for all $1 \le i \le n$.*

Proof. Note that in odd rounds, all robots with an odd index move in negative x-direction and robots with an even index in positive x-direction. In even rounds, the robots move in the respective other direction. This is obvious for the first round (all robots positioned above v_0 move above v_{n+1} and the other way round). Moreover, if for a robot v_i both neighbors moved in the same direction (or did not move at all) in round t, v_i's target point also moved in this direction in round t and thus v_i will equally move in this direction in round $t+1$. So for each robot v_i, both neighbors always move in the same direction. Therefore,

$$\begin{aligned} d_i(t) &= \left|\frac{1}{2}v_{i-1}(t) + \frac{1}{2}v_{i+1}(t) - (\frac{1}{2}v_{i-1}(t-1) + \frac{1}{2}v_{i+1}(t-1))\right| \\ &= \left|\frac{1}{2}v_{i-1}(t) - \frac{1}{2}v_{i-1}(t-1) + \frac{1}{2}v_{i+1}(t) - \frac{1}{2}v_{i+1}(t-1)\right| \\ &= \left|\frac{1}{2}v_{i-1}(t) - \frac{1}{2}v_{i-1}(t-1)\right| + \left|\frac{1}{2}v_{i+1}(t) - \frac{1}{2}v_{i+1}(t-1)\right| \\ &= \frac{1}{2}d_{i-1}(t) + \frac{1}{2}d_{i+1}(t). \end{aligned}$$

The third equality holds, because both neighbors of each robot always move in the same direction. ■

Two further lemmas are necessary before we can show the lower bound on the traveled distance. The first one is a technical lemma which will be helpful later.

Lemma 4.17. *For the tower start configuration, the x-distance traveled by each robot is monotonically decreasing with time.*

Proof. We prove $d_i(t) \leq d_i(t-1)$ for all robots v_i and all $t \in \mathbb{N}$ by induction over t.
Base case: Each robot travels a x-distance of d in the first round. Due to the construction of the configuration, no robot can ever move further than a x-distance of d, and so in the second round, each robot travels at most as far in x-direction as in the first round.
Inductive step: Consider an arbitrary robot v_i in round t.

$$\begin{aligned} d_i(t) &= \frac{1}{2}d_{i-1}(t-1) + \frac{1}{2}d_{i+1}(t-1) \\ &\leq \frac{1}{2}d_{i-1}(t-2) + \frac{1}{2}d_{i+1}(t-2) \\ &= d_i(t-1) \end{aligned}$$

The inequality holds because of the induction hypothesis. ■

The next lemma forms the basis of the lower bound on the traveled distance, since this is only possible if the number of rounds is also large.

Lemma 4.18. *For the tower start configuration, $k = \Omega(n^2)$ rounds are necessary until every robot has a y-distance of at most 1 from its end position.*

Proof. We use $Y(t) = \sum_{i=1}^{n} |Y_i(t)|$ again as our progress measure, where $Y_i(t)$ is the y-distance of robot v_i to its end position in round t. For the tower start configuration, $Y(0) = 2\sum_{i=1}^{n/2} \frac{6}{n} \cdot i = \frac{3}{2}n + 3 \geq \frac{3}{2}n$. According to Lemma 4.9, $Y(t)$ decreases by at most $6/n$ per round and thus, after t rounds

$$Y(t) \geq Y(0) - t \cdot \frac{6}{n} \geq \frac{3}{2}n - t \cdot \frac{6}{n}.$$

So there is one robot i for which

$$|Y_i(t)| \geq \frac{3/2 \cdot n - 6/n \cdot t}{n} = \frac{3}{2} - t \cdot \frac{6}{n^2}.$$

Therefore $|Y_i(t)| \leq 1$ only for $t \in \Omega(n^2)$. ■

Now we can finally show that there is a robot which has to travel a distance of $\Omega(n^2)$ when starting in a tower configuration.

Lemma 4.19. *Consider the tower start configuration. If $\delta \geq \frac{7}{n}$, there is a robot r such that the distance traveled by r until all robots are at most in distance 1 from their end positions is at least $\frac{\delta}{56}n^2 + \frac{\delta}{28}n$.*

Proof. Let k be the last round before all robots are in distance at most 1 from their end position, and define $d_i := \sum_{t=1}^{k} d_i(t)$. According to Lemma 4.16, $\sum_{t=2}^{k} d_i(t) = \frac{1}{2}\sum_{t=1}^{k-1} d_{i-1}(t) + \frac{1}{2}\sum_{t=1}^{k-1} d_{i+1}(t)$. Since in the first round every robot travels in x-direction a distance of d, $\sum_{t=1}^{k} d_i(t) = d + \sum_{t=2}^{k} d_i(t)$.

If there is a robot in round k which moves a distance of at least $\frac{d}{2}$, since the distance traveled is monotonically decreasing (Lemma 4.17), it is obvious that this robot moves a distance of at least $\frac{d}{2} \cdot n^2 = \frac{\delta}{14}n^2 \geq \frac{\delta}{56}n^2 + \frac{\delta}{28}n$ and we are done. So assume that in the last round every robot moves at most $\frac{d}{2}$. This yields $\sum_{t=1}^{k} d_i(t) \leq \frac{d}{2} + \sum_{t=1}^{k-1} d_i(t)$. We can combine this to obtain

$$\begin{aligned} d_i &= \sum_{t=1}^{k} d_i(t) = d + \sum_{t=2}^{k} d_i(t) = d + \frac{1}{2}\sum_{t=1}^{k-1} d_{i-1}(t) + \frac{1}{2}\sum_{t=1}^{k-1} d_{i+1}(t) && (4.4)\\ &\geq d + \frac{1}{2}\sum_{t=1}^{k} d_{i-1}(t) - \frac{d}{4} + \frac{1}{2}\sum_{t=1}^{k} d_{i+1}(t) - \frac{d}{4} && (4.5)\\ &= \frac{d}{2} + \frac{1}{2}d_{i-1} + \frac{1}{2}d_{i+1}. && (4.6) \end{aligned}$$

Similar to Proposition 4.14, we can use this to obtain lower bounds for the movement of each robot which only depends on the neighbor which is further apart from a station:

Proposition 4.20. *1. For $1 \leq i < \frac{n}{2}$, $d_i \geq \frac{d}{2}i + \frac{i}{i+1}d_{i+1}$*

2. For $\frac{n}{2}+1 < i \leq n$, $d_i \geq \frac{d}{2} + \frac{n-i}{n-i+1}d_{i-1}$.

Proof. We will show for $1 \leq i < \frac{n}{2} : d_i \geq \frac{\delta}{2}i + \frac{i}{i+1}d_{i+1}$ via induction. The base case is true, since we know $d_i \geq \frac{d}{2} + \frac{1}{2}d_0 + \frac{1}{2}d_2$ and $d_0 = 0$. For the inductive step, it holds that

$$\begin{aligned} d_{i+1} &\geq \frac{d}{2} + \frac{1}{2}d_i + \frac{1}{2}d_{i+2} \\ &\geq \frac{d}{2} + \frac{d}{4}i + \frac{i}{2\,(i+1)}d_{i+1} + \frac{1}{2}d_{i+2}. \\ \Leftrightarrow \left(1 - \frac{i}{2\,(i+1)}\right)d_{i+1} &\geq \frac{i+2}{4}d + \frac{1}{2}d_{i+2} \\ \Leftrightarrow d_{i+1} &\geq \frac{i+1}{2}d + \frac{i+1}{i+2}d_{i+2}. \end{aligned}$$

The first inequality follows from (4.6) in Lemma 4.19 and the second from the induction hypothesis. Additionally, because of the symmetry, we know that for $\frac{n}{2}+1 < i \leq n : d_i \geq \frac{d}{2}(n-i) + \frac{n-i}{n-i+1}d_{i-1}$. ■

Now fix one of the robots at the top: we show that this robot has to travel a long distance, plugging in the results from Proposition 4.20. The proof is similar to the proof of Proposition 4.15:

Proposition 4.21. $d_{\frac{n}{2}} \geq \frac{dn^2}{8} + \frac{dn}{4}$

Proof. Using our previous results, we may write

$$\begin{aligned} & d_{\frac{n}{2}} \geq \frac{d}{2} + \frac{1}{2}d_{\frac{n}{2}-1} + \frac{1}{2}d_{\frac{n}{2}+1} = \frac{d}{2} + \frac{1}{2}d_{\frac{n}{2}-1} + \frac{1}{2}d_{\frac{n}{2}} \\ \Leftrightarrow\ & \frac{1}{2}d_{\frac{n}{2}} \geq \frac{d}{2} + \frac{1}{2}d_{\frac{n}{2}-1} \\ \Leftrightarrow\ & d_{\frac{n}{2}} \geq d + \left(\frac{n}{2} - 1\right)\frac{d}{2} + \frac{\frac{n}{2}-1}{\frac{n}{2}}d_{\frac{n}{2}} \geq \frac{n}{4}d + \frac{d}{2} + \frac{n-2}{n}d_{\frac{n}{2}} \\ \Leftrightarrow\ & \frac{2}{n}d_{\frac{n}{2}} \geq \frac{n}{4}d + \frac{d}{2} \\ \Leftrightarrow\ & d_{\frac{n}{2}} \geq \frac{n^2}{8}d + \frac{dn}{4}, \end{aligned}$$

which in turn yields the proposition. ■

Since $d = \frac{\delta}{7}$, this yields the lemma. ■

Conclusion

We can now combine the obtained lemmas to prove the claim of the total distance traveled for the discrete strategies.

Theorem 4.22. *For a worst-case start configuration, the maximum distance traveled by a robot is $\Theta(\delta n^2 + n)$, when using δ-GTM.*

Proof. According to Lemma 4.5 and Lemma 4.7, the first phase takes $\Theta(\frac{n}{\delta})$ rounds. Due to the definition of the first phase, there exists a robot which moves a distance of δ in each round of the first phase, while all other robots travel at most a distance of δ in each round. Thus, this robot travels a distance of $\Theta(n)$, all others of $O(n)$. Since the distance traveled in the second phase is $\Theta(\delta n^2)$ according to Lemma 4.13 and Lemma 4.19 for $\delta \geq \frac{7}{n}$, we get an overall traveled distance D of $\max\{\Theta(n), \Theta(\delta n^2)\} \leq D \leq \mathcal{O}(\delta n^2 + n)$ and therefore $D \in \Theta(\delta n^2 + n)$ (note that the distance traveled in the second phase is longer only if $\delta \geq \frac{7}{n}$). ■

Corollary 4.23. *For a worst-case start configuration, when using 1-GTM, the maximum distance traveled by a robot is $\Theta(n^2)$.*

To interpret the results, we can see that similar to the number of rounds, the traveled distance is longer in the first phase if $\delta \in \mathcal{O}(\frac{1}{n})$. If $\delta \in \Omega(\frac{1}{n})$, the traveled distance in the second phase phase is longer. Again, for worst case instances no optimal global strategy can be better than $\Theta(n)$ and thus for $\delta \in \mathcal{O}(\frac{1}{n})$, the traveled distance is asymptotically optimal compared to a global strategy. While the number of rounds are minimized for $\delta \in \Omega(\frac{1}{n \log n})$, the maximum traveled distance is minimized for $\delta \in \mathcal{O}(\frac{1}{n})$. So the next theorem follows.

Theorem 4.24. *For $\delta \in \Theta(\frac{1}{n})$, the number of rounds is $\mathcal{O}(n^2 \log n)$ and the traveled distance is $\mathcal{O}(n)$. Thus both quality measures are minimized for this strategy.*

4.2 The continuous Go-To-The-Middle strategy

We have seen in the last section that choosing $\delta \in \Theta(1/n)$ minimizes the movement distance as well as the number of rounds. Nevertheless, δ can not always be chosen freely, but sometimes robots are able to sense the positions of their neighbors continuously, while it costs a lot of energy to move the robots over some distance. In this scenario, the continuous time model is close to practical applications [NPR+03]. Therefore this section is dedicated to the continuous GO-TO-THE-MIDDLE strategy. We will show that similar to MOVE-ON-BISECTOR, the distance moved by robots using Continuous-GTM and therefore also the time needed until all robots are on the target line is asymptotically optimal for worst-case instances.

For bounding the maximum traveled distance, we start with the easy observation that no strategy can be faster than $\Omega(n)$ in the worst case.

Lemma 4.25. *There are start configurations for which the maximum distance traveled by a robot is $\Omega(n)$ using an optimal algorithm.*

Proof. When starting with a configuration in a triangular shape, such that each robots is in a constant y-distance from each neighbor (see Fig. 4.3), the middle robot is in distance $\Omega(n)$ of the line between the two stations, and thus it must travel at least this distance to reach its end position. ■

The following theorem shows that Continuous-GTM reaches this bound: it is asymptotically optimal compared to a global algorithm.

Theorem 4.26. *When using Continuous-GTM, the maximum distance traveled by a robot is $\Theta(n)$ for a worst-case start configuration.*

Proof. The lower bound follows from Lemma 4.25. So we need to show that when using Continuous-GTM, no robot moves more than for a distance of $\mathcal{O}(n)$. According to Lemma 4.7, in the discrete setting the first phase takes $c \cdot \frac{n}{\delta}$ rounds for some constant c. Since in each round the robots move at most a distance of δ, the distance traveled in the first phase is bounded by $c\frac{n}{\delta} \cdot \delta \leq cn = \mathcal{O}(n)$. Let us now consider the limit $\delta \to 0$, yielding Continuous-GTM. Since the upper bound on the traveled distance in the first phase is independent of δ, it remains valid. On the other hand, Continuous-GTM does not have a second phase, since the robots reach their end positions exactly when the last robot reaches its target point. Therefore, the overall distance traveled by the robots is $\mathcal{O}(n)$. ■

According to Lemma 4.25 and Theorem 4.26, Continuous-GTM is asymptotically optimal regarding the traveled distance.

We have seen in Chapters 3 and 4 that the robot chain problem can be solved efficiently in terms of the number of rounds as well as the maximum traveled distance by local algorithms and very simple robots. If only considering the movement distance, Move-On-Bisector is optimal for worst-case instances and $\log n$-competitive for any kind of instances with a small d when compared to an optimal global algorithm. Combining both quality measures, δ-GTM shows that when choosing δ correctly, the number of rounds of the original Go-To-The-Middle-strategy can be preserved while the movement distance is reduced to $O(n)$, which is again optimal for worst-case instances.

The analyses for the robot chain problem rely heavily on one structural property of the problem: That each robot has exactly two neighbors, which do not change during the execution of the algorithm. This is not possible for the gathering problem, which we will tackle now in Chapters 5 to 8. Here, each robot can have arbitrarily many neighbors, and the neighbors can change. We will therefore have to use different techniques for the analysis.

Chapter 5

The Gathering Problem

5.1 Introduction

As a second robot formation problem, we study a classic mobile network problem, the robot-gathering problem, under locality. In the beginning, n robots with a limited viewing range are placed in the plane, such that their visibility graph is connected. The goal is to gather them in a not predefined point. It is known for 16 years that this is possible in finite time in the synchronous LCM model, which we also used for the robot chain problem. In their seminal paper, Ando, Suzuki, and Yamashita [ASY95, AOSY99] presented an algorithm that gathers the robots. In each round, every robot simply moves to the center of the smallest enclosing circle of the robots in its viewing range, only constrained by the condition that robots must not lose visibility to their neighboring robots. As Ando, Suzuki and Yamashita proved, this approach works, and the robots eventually meet. Since then it has been an open question how many rounds are required to achieve such a gathering. Chazelle [Cha09] showed that similar processes may have an exponential behavior. Therefore, our main focus are efficient algorithms for the gathering problem with respect to the number of rounds.

Outline Chapters 6 and 7 are devoted to two runtime analyses of two different gathering algorithms. The first one, called MoveInCH, is the first known local algorithm with a proven runtime bound. It is designed for its analysis and is therefore rather complicated. It works in a standard asynchronous time model, in which no two neighboring robots are active at the same time, and achieves gathering in expected $O(n^2)$ asynchronous rounds, if the order of activation is at random (see Section 5.2 for a detailed model description). A round in this asynchronous setting ends as soon as all robots have been active at least once. The idea of the algorithm is that each robot tries to reduce the area of the convex hull of the robot positions. To achieve this, the robots need the ability to not only determine a target point for themselves, but also for their neighbors. The LCM model therefore needs to be adapted: Between the Compute- and the Move-Operation, a Communication-Operation is introduced. In this Communication-Operation, a robot can communicate the computed target points to its neighbors, such that all robots can move to their target point in the following Move-Operation. Note that since the robots are oblivious, the Communication-Operation does not give the robots the ability to gather global information. We will describe this algorithm in detail in Chapter 6.

MOVEINCH is a local algorithm and achieves gathering very quickly. Nevertheless, it is desirable to have an efficient and simple algorithm which works in the synchronous model and does not need the Communication-Operation. In Chapter 7 we show that this is indeed possible. In particular, we show that the algorithm by Ando, Suzuki and Yamashita gathers the robots in $\Theta(n^2)$ synchronous rounds.

Knowing that gathering the robots is possible in only $\Theta(n^2)$ rounds, we then turn towards the movement distance which is needed for gathering. In Chapter 8, we introduce a variant of the MOVE-ON-BISECTOR strategy (see Section 2.3.1 and Chapter 3) and use techniques similar to those from Chapter 3 to analyze the maximum traveled distance. Like for MOVE-ON-BISECTOR, we will see that the maximum traveled distance is small even compared to an optimal global algorithm.

We will now proceed with a formal problem description, give details about the models and the notation and describe our quality measures. The algorithms will be described in the respective chapters.

5.2 Problem description and notation

Given a set $\mathcal{R}$ of n robots $v_1, \ldots, v_n$ in the Euclidean plane, the goal is to gather all robots in one point, which is not determined in advance. Contrary to the robot chain problem, we can now control the movement of all robots. We are again constrained by a limited *viewing range*, which is constant but depends on the algorithm. Moreover, two robots are connected, if they are within distance 1 of each other. We call this distance the *connection range*. The notion of the connection range induces a unit disk graph, the *connection graph* $G_t = (\mathcal{R}, E_t)$, where $(v_i, v_j) \in E_t$ iff v_i and v_j are mutually connected at time t, i.e. they are within distance 1 of each other. In order to be able to achieve a gathering of the robots, we need to assume G_0 to be connected. Our algorithms will keep G_t connected at all times in order to make sure that the robots do not split into several groups. The algorithm MOVEINCH, which is introduced and analyzed in Chapter 6, uses a viewing range of 2, such that the robots can see twice as far as the connection range. The algorithm by Ando, Suzuki and Yamashita, which is analyzed in Chapter 7, only needs a viewing range of 1.

Additionally, our robots are oblivious, which means that they do not have a memory, they do not use a common coordinate system, and they do not have IDs. For the algorithm by Ando, Suzuki and Yamashita they also do not need to communicate. MoveInCH uses a little bit of communication: robots compute target points for their neighbors and need to tell them about it. Note that this does not allow the robots to compute a global solution, since the robots are oblivious and forget everything they have learned when they are active next.

Again we abstract from technical issues. That is, robots can observe their neighborhood accurately, they can compute geometric properties and they can share a position with other robots.

Note that if the robots had full visibility, the problem would be trivial in the synchronous LCM model as all robots could compute the unique center of the smallest enclosing circle (SEC) of all robots, and then concurrently move there, finishing in one single round. Similarly, unique IDs or a common coordinate system would make the problem much easier, since the robots could agree to meet at the position of the robot with the smallest ID or which is positioned in the lower left corner.

Time and movement models

As for the robot chain problem, our time models are based on the discrete LCM model. For the gathering problem, we distinguish two different types of LCM models.

The algorithm by Ando, Suzuki and Yamashita operates in the classical *synchronous LCM model* as used in Chapter 4, such that all robots perform their LCM cycle at the same time. We will investigate this algorithm in Chapter 7.

MoveInCH uses a standard asynchronous variant of the LCM model. Only one robot is active at a time, and when active, the robot performs a complete LCM cycle. We call this a *time step.* 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. This allows to abstract from symmetry breaking issues. Usually, the analysis of robotic strategies in this model is done assuming activation of robots in worst case order in each round.

In this thesis we assume weaker models for activation: In the *random order model*, we assume that, in each time step, a randomly, uniformly chosen robot becomes active. The choices in different time steps are independent.

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

These time models are used for the analysis of MOVEINCH. 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 awaken robot becomes active, if no other robot in its connection range is awaken. 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 6.3, 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 (the *local activation model*). It employs a distributed protocol for handling interferences which is tailored to MOVEINCH. We will prove a $\mathcal{O}(n^2)$ bound for the expected number of time steps instead of rounds in this model.

If we refer to a *time* t, we refer to either the end of a time step in the asynchronous model or the end of a round in the synchronous model. In the synchronous model, we use the notion of a round interchangeably with a time step.

Notation

Given a time step or a round $t \in \mathbb{N}_0$, the position of robot v_i at this time is denoted by $v_i(t) \in \mathbb{R}^2$. As for the robot chain problem, we call the positions $v_1(t), \ldots, v_n(t)$ of the robots at the end of round t the *configuration* at time t. The configuration at time 0 is called *start configuration*. When clear from the context, we will sometimes also refer to a robot v_i's position by v_i. $d(v_i(t), v_j(t))$ describes the Euclidean distance between the two robots v_i and v_j. If the round or time step is clear from the context, we will sometimes denote this distance by $d(v_i, v_j)$.

Two robots v_i and v_j can see each other, if $d(v_i, v_j) \leq C$, where we call v_i and v_j *neighbors* and the distance C the *viewing range* of the robots. C depends on the algorithm. The set of all neighbors of a robot v_i – its *neighborhood* – at time t is denoted as $N_t(v_i)$ or just $N(v_i)$ when the time is clear from the context.

Part of the analysis is based on the convex hull of the robot positions, to which we will also refer by the convex hull of these 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* $C_i(t)$ of a robot v_i at time t. The local convex hull is the convex hull of all robots which are within viewing range of v_i at time t.

As soon as two robots share the same position, our algorithms will keep the robots together. So we say that two robots *fuse* when they share the same position for the first time. Concerning the asynchronous random round models, they now act as one robot, that is, their probability to be activated in a time step t is equal to the probability of one single robot.

5.3 Quality Measures

While the robot chain problem was already analyzed well regarding the number of rounds needed to solve the problem, this is not the case for gathering. Thus, our main focus in the next chapters is on the number of rounds until the robots have gathered. Chapter 8 deals with the maximum movement distance needed for gathering.

For the synchronous time models which we used so far, the number of rounds can be interpreted as the number of neighborhood observations per robot, which can be relevant in terms of energy. This is also the case in the asynchronous random permutation model, since each robot is activated exactly once per round. That is, for the asynchronous random permutation model, the number of neighborhood observations is $\mathcal{O}(n^2)$ for both presented algorithms. But for the random order model, this can be different: Some robots may be active several times before a round ends. Here, a round will in expectation take $\mathcal{O}(n \log n)$ time steps (coupon collector problem), and a fixed robot will be active $\mathcal{O}(\log n)$ times in expectation. Thus, if an algorithm takes X rounds in the random order model, a fixed robot will be active $X \log n$ times in expectation. For the algorithm MoveInCH, which we analyze in this model, this results in an expected number of neighborhood observations of $O(n^2 \log n)$ per robot. Concerning the local activation model, since we prove a runtime of $O(n^2)$ time steps here, the number of neighborhood observations is bounded by $O(n^2)$.

For the movement distance, which is analyzed for a variant of Move-On-Bisector in Chapter 8, we again bound the maximum traveled distance, the maximum taken over all robots. As the robots gather in one point and the algorithm is deterministic, this distance is fixed for a fixed start configuration.

Chapter 6

A first gathering algorithm

This chapter presents the first algorithm for gathering with known runtime bounds. In order to achieve an analysis, the algorithm is rather complicated and the robots need additional capabilities compared to those needed for Go-To-The-Middle or the algorithm which is presented in Chapter 7. In particular, the robots have a viewing range of 2 instead of 1, while the Unit Disk Graph of the robots is still connected with respect to the connection range, which is 1. Thus, the robots can look twice as far. Moreover, the robots do not only compute target points for themselves, but also for their neighbors. As described in Section 5.1, they need a Communication-Operation between the Compute- and the Move-Operation of the LCM model in order to communicate the computed target point to their neighbors. But because the robots are still oblivious, this does not allow them to compute a global solution.

We will now first describe the algorithm in Section 6.1, then we analyze it in Section 6.2. In Section 6.3 we present the local activation model, and show that $O(n^2)$ time steps suffice in this model to achieve the gathering with MoveInCH. This shows that a global activation of the robot is not necessary. Finally we give a short conclusion in Section 6.4.

6.1 The algorithm MoveInCH

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 robots. 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 MoveInCH. It is executed by robot v_i at the time t in which it is active (Algorithm 1). See Fig. 6.1 for an illustration of

Algorithm 1 MoveInCH: The algorithm for robot v_i at time t:

1: Compute the sets A_i and B_i of the robots within the viewing resp. connection range of v_i. Let C_i denote the convex hull of A_i.

2: (Termination) If $A_i = B_i$ (i.e. no robots from A_i have distance between 1 and 2 to v_i), then move all robots from B_i to the position $v_i(t)$.

3: Else (B_i is a proper subset of A_i)

3.a: (Fuse) If the positions in B_i can be rearranged such that the resulting new set A_i' is still contained in C_i, is still connected, and at least two robots share the same position (are fused), perform this rearrangement. Fused robots will alway have the same position from now on.

3.b: (Reduction) If a fusion is not possible and v_i lies on the boundary of C_i, then do the following:

1. Compute the two first intersections of the boundary of C_i with the boundary of v_i's connection range if started from $v_i(t)$ in clockwise/ counter-clockwise direction. (Note that these are the intersections which are in maximum distance to each other.)
2. Compute the line segment l between these intersections.
3. Move all robots on v_i's side of l to their respective closest point on l.

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.

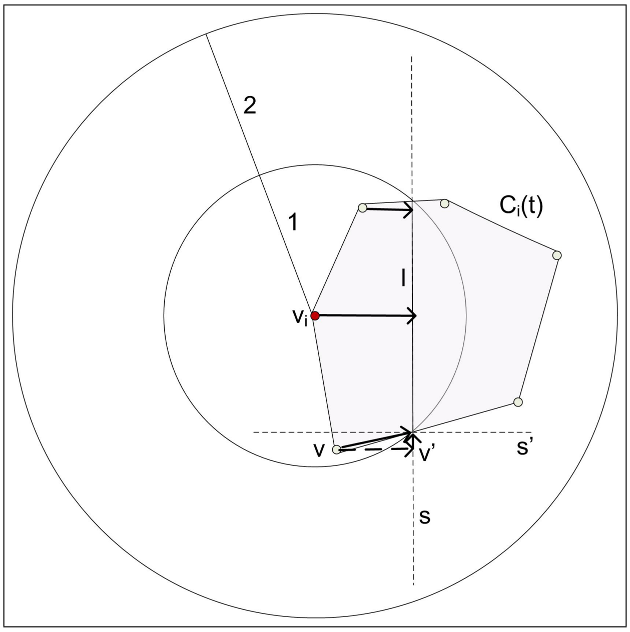


Figure 6.1: Illustration of step 3b of the algorithm and its correctness

6.2 Analysis of MoveInCH

This section is dedicated to the analysis of the correctness and runtime of the algorithm MoveInCH. 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 6.1. *If the network is connected before a robot v_i executes the algorithm, it is still connected afterwards.*

Proof. If the action in step 2 of the algorithm is executed, all robots which are in the viewing range of v_i, are also in its connection range. 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 v_i executes the action in step 3a of the algorithm, the robots in v_i's viewing range stay connected by definition. Moreover, since only robots within v_i's connection range are moved, edges of the unit disk graph G_t ending outside v_i's viewing range are not affected.

Now let v_i be a robot executing the action in step 3b of the algorithm in time step t. Since again only robots within v_i's connection range are moved, we only need to prove that all robots in the local convex hull $C_i(t)$ 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 Fig. 6.1 for an illustration. Let R_1 be the subset which contains v_i, 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 point by projecting it to another straight line s': s' is orthogonal to l and intersects l in v's target point (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. ■

Corollary 6.2. *If Step 2 of the algorithm (termination) is executed, the algorithm has gathered all robots in one position.*

Note that in rounds without a fusion, 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 6.3. *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 v_i of the convex hull moves itself and neighboring robots to the line segment l, l is completely inside the local convex hull of v_i and therefore again no robot leaves the global convex hull. ■

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

Lemma 6.4. *If $|B_i| > 16$ for a robot v_i, then Step 3a of the algorithm (Fusion) is always possible.*

Proof. Insert a grid with step width 1 into the intersection of v_i's viewing range and $C_i(t)$. It is always possible to insert such a grid which has at most 16 points. If there are more robots in v_i'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 v_i's viewing range stays connected and that no robot leaves $C_i(t)$. ■

Progress in rounds without fusion 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 a fusion. 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 6.8). 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 6.6). 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 $v_1, ..., v_m$.

- β_i^* is the internal angle of the global convex hull at vertex robot v_i at the beginning of the round.

Now consider a vertex robot v_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 v_i's connection range if started from $v_i(t)$ in clockwise/counterclockwise direction (the intersections which are in maximal distance to each other). Let T denote the triangle with the vertices $v_i(t)$, p_1 and p_2. Then β_i is the internal angle of T in vertex $v_i(t)$.

- Let p'_1 and p'_2 denote the intersections of the *local* convex hull C_i and the boundary of v_i's connection range if started from $v_i(t)$ in clockwise/ counter-clockwise direction (the intersections which are in maximal distance to each other). Let T^* denote the triangle with the vertices $v_i(t)$, p'_1 and p'_2. Then α_i is the internal angle of T^* in vertex $v_i(t)$.

Figure 6.2 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.

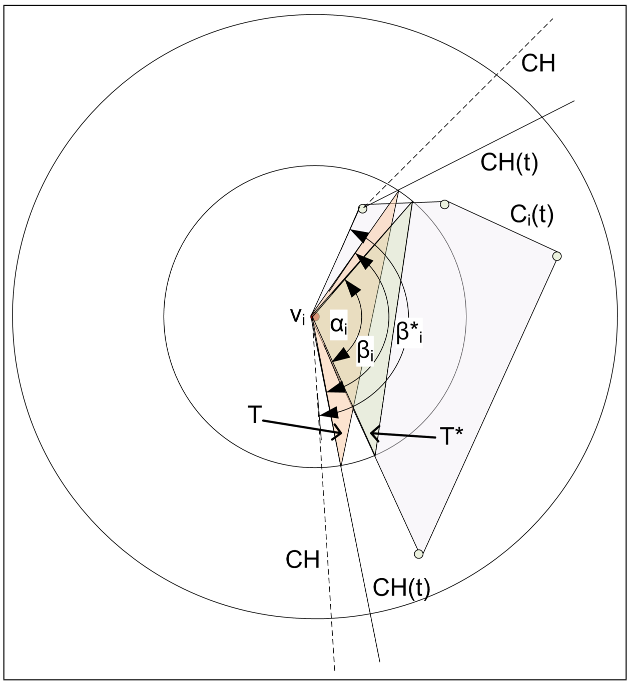


Figure 6.2: Angles used in this subsection. CH indicates the global convex hull at the beginning of the round. β_i and α_i are internal angles of the triangles T and T^* at the first time step in which v_i turns active in the round, β_i^* is the internal angle of the global convex hull at the beginning of the round.

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 6.5. *Consider a fixed round in which no robots are fused. Then α_i is greater than $\frac{\pi}{3}$ for all robots v_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 v_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 $p_{v_i}(t)$. See Fig. 6.3 for an illustration. v_i can be moved to the point inside the local convex hull closest to the point shown as p in Fig. 6.3. Moreover, because G_t is always connected, there must have been at least one robot in the connection range of v_i. All these robots can now fuse with v_i. Afterwards, no robots remain in the old connection range of v_i and thus the robots from $C_i(t)$ are connected. ■

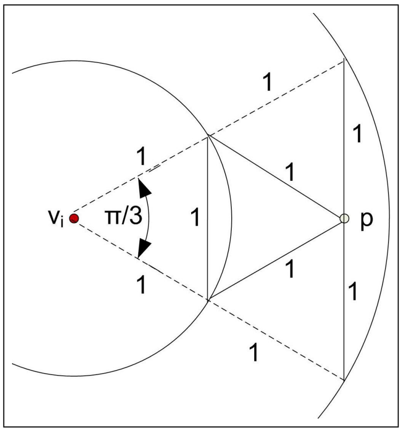


Figure 6.3: Illustration of a position from which all neighbors are in connection range. The indicated sector of the circle must contain the local convex hull, if $\alpha_i \leq \frac{\pi}{3}$.

Proposition 6.6. *Consider a fixed round in which no robots are fused, and a robot v_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 $C_i(t)$, no point of l or the circular segment defined by l and the connection range of v_i can lie strictly inside of T (see Fig. 6.4 for an illustration). As robot v_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 6.5: According to this proposition, α_i is at least $\frac{\pi}{3}$, giving that $\sin(\frac{\alpha_i}{2}) \geq \frac{1}{2}$. ■

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 6.4 states an upper bound for c of 16.

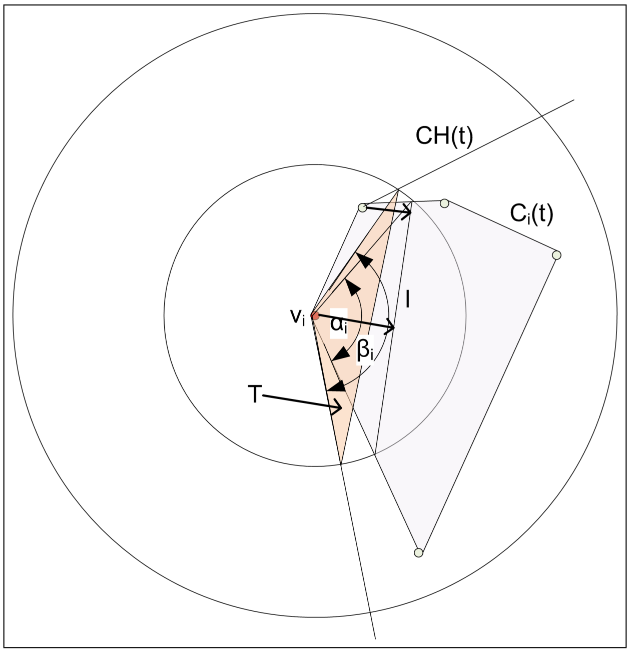


Figure 6.4: Illustration of the proof of Proposition 6.6

Lemma 6.7. *The probability that a vertex robot v_i is not moved by the activation of another robot prior to its own activation during the same round is at least $\frac{1}{c+1}$.*

Proof. At the moment t in which it is active, a robot v_i can have at most c neighbors (in its connection range), since otherwise robots could be fused. Let $c' \leq c$ be this number of neighbors. Then the robot v_i can also have at most c' neighbors in all time steps of the current round before t, because once neighbors, robots stay connected at least until the next fusion. If during the round, v_i becomes active before its c' neighbors at time t, it cannot have been moved by the activation of another robot prior to its own activation. Therefore, we lower bound the probability that this event occurs. In the random order model, it is at least

$$\sum_{t=0}^{\infty} \frac{1}{n}(1-\frac{c'+1}{n})^t = \frac{1}{n}\frac{1}{1-(1-\frac{c'+1}{n})} \quad = \quad \frac{1}{c'+1} \geq \frac{1}{c+1}.$$

In the random permutation model, let C' be the set of the robots v_i and its c' neighbors when v_i is active, $|C'| = \frac{1}{c'+1}$. Then the probability that v_i is the first robot out of C' in the random permutation that determines the order of activation of this round is $\frac{1}{c'+1} \geq \frac{1}{c+1}$. ■

Lemma 6.8. *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 $\frac{1}{c+1}$.*

Proof. Let v_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 v_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 v_i truncates the global convex hull by $\frac{1}{2}\cos(\frac{\beta_i}{2})$ (Proposition 6.6), we need a relation between the internal angle of CH in robot v_i in the beginning of the round (β_i^*) and β_i, the internal angle in robot v_i of the triangle T at the beginning of the first time step t in which v_i is active.

Note that v_i may have neighbors which were moved by other robots before. Consider the case that v_i is active before any other robot in its connection range. This means that v_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 v_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 v_i at the beginning of time step t, and thus $\beta_i^* \geq \beta_i$, if v_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 v_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 v_i in the round.

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

The equality follows from Lemma 6.7. 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:

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

■

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 6.3). 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 6.9. *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 v_i 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.

According to Corollary 6.2, 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 $n-1$ robot 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_t to be connected. Since according to Lemma 6.8 the area of the convex hull is truncated in expectation by a constant in a round without a fusion and since the area of the global convex hull never increases (Lemma 6.3), there can be at most $\mathcal{O}(n^2)$ rounds in expectation without a fusion. Summing up the number of rounds for each operation leads to the desired bound. ∎

6.3 A local random activation 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 v_i first computes the size b_i of B_i, the set of robots within its connection range. Then it wakes up with probability $\frac{1}{\max(c,b_i)}$, where c is the maximum number of neighbors a robot can have without fusing robots. It becomes active if no other robot v_j in B_i with a smaller b_j woke up. If v_i 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_i, followed by just one step for parallel executions of our local gathering algorithm.

Theorem 6.10. *Our local gathering algorithm needs expected $O(n^2)$ time steps in the local activation model.*

Proof. Consider a time step.

If no robot fusion is possible, each robot v_i wakes up with probability $\frac{1}{c}$, and becomes active with probability $p \geq \frac{1}{c}(1-\frac{1}{c})^{c-1}$. As p is constant, Lemma 6.8 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 a robot fusion suffice.

If a fusion is possible, $b_i > c$ holds for some robot v_i. v_i becomes active with probability

$$P_i = \frac{1}{b_i} \cdot Prob(\text{none of } v_i\text{'s neighbors wake up}) = \frac{1}{b_i} \cdot \prod_{j=1}^{b_i}(1 - \frac{1}{max\{c, b_j\}}).$$

Now consider the neighbors of v_i. They are all inside the circle C with radius 1 around v_i. This area can be covered with a constant number of circles C_j with radius 1/2. All robots in one circle can all mutually see each other. Let $|C_j|$ be the number of robots in circle C_j. Let furthermore c' be the number of circles which contain exactly one robot and c'' the number of circles which contain at least two robots.

First consider those robots which are alone in their circle. Since the unit disk graph of the robots is connected, each of those robots has at least one neighbor and thus their probability to wake up is at most $1/2$. Each other robot has at least as many neighbors as there are robots in its circle, and thus for such a robot v_k in circle C_j, its probability not to wake up is at least $1 - \frac{1}{|C_j|}$. We can use this to rewrite the probability that v_i becomes active:

$$P_i \geq \frac{1}{b_i} \cdot \left(\frac{1}{2}\right)^{c'} \cdot \prod_{j=1}^{c''} (1 - \frac{1}{|C_j|})^{|C_j|}.$$

Note that robots may be counted several times, as they can be in more than one circle, but this only reduces the computed probability. Finally we use that $b_i \leq n$ and $(1 - \frac{1}{|C_j|})^{|C_j|} \geq \frac{1}{4}$ for $|C_j| \geq 2$:

$$P_i \geq \frac{1}{n} \cdot \left(\frac{1}{2}\right)^{c'} \cdot \left(\frac{1}{4}\right)^{c''}$$

Thus expected $O(n^2)$ time steps in which a fusion is possible suffice to perform all at most $n-1$ many fusions. ■

6.4 Conclusion

We have shown that using the MOVEINCH algorithm with one of the described random round models, $\mathcal{O}(n^2)$ rounds respectively time steps are required in expectation to achieve gathering. Regarding the number of neighborhood observations, it is in expectation $\mathcal{O}(n^2)$ per robot in the random permutation model and the local activation model, where each robot is active at most once per round or time step. In the random activation model, a robot is active $O(\log n)$ times in expectation in each round. This results in an expected number of neighborhood observations of $\mathcal{O}(n^2 \log n)$ per robot. Note that these bounds do not directly give an upper bound for the movement distance, since when using the MOVEINCH algorithm, robots may also move when not active.

This algorithm shows that it is possible to efficiently gather the robots in one point using only local information. Its drawback is that it is rather complicated, that it cannot be executed in a strictly synchronous time model, that the robots need to compute target points for their neighbors and that a randomized round model is needed for the runtime bound. We will show in the next chapter that this is not inherent to the problem: Local gathering is also possible efficiently with very simple robots in a synchronous model.

Chapter 7

An improved gathering algorithm

In Chapter 6 we have seen a first local algorithm for the gathering problem with runtime bounds. In this chapter we focus on a very simple and intuitive algorithm. The idea of the algorithm, which was originally presented by Ando, Suzuki and Yamashita [ASY95, AOSY99], is to move all robots towards the center of the smallest enclosing circle (*SEC*) of their neighbors, stopping them as soon as they might lose connectivity to one of their neighbors. The smallest enclosing circle is the point which minimizes the maximum distance to its neighbors. This algorithm is executed synchronously, and contrary to MoveInCH, no communication is needed. Ando, Suzuki and Yamashita showed that this simple algorithm gathers the robots in finite time, but so far runtime bounds were unknown. We show that $\mathcal{O}(n^2)$ synchronous rounds suffice to gather the robots in one point. This algorithm is therefore as fast as MoveInCH, but overcomes its drawbacks.

Next, we give a formal description of the algorithm (Section 7.1). In Section 7.2, we show that there is a configuration for which $\Omega(n^2)$ rounds are necessary to gather the robots, before we proceed to the main result in Section 7.3: Gathering is achieved in $\mathcal{O}(n^2)$ rounds.

7.1 The algorithm Go-To-The-Center

The algorithm is based on the smallest enclosing circle (*SEC*) of a point set $\mathcal{P}$ (which are robot positions in our context). Its center is the point that minimizes the maximum distance to any point in $\mathcal{P}$.

The GTC-algorithm, which was first introduced in [ASY95], works as follows. First, v_i computes its *target point* $c_i(t)$, which is the center of the smallest enclosing circle around itself and its neighbors. Because the connectivity of the unit disk graph could break if all robots would move to their target point, a second phase

Algorithm 2 GO-TO-THE-CENTER: The algorithm for robot v_i in round t

$\{$*compute target point*$\}$
$\mathcal{R}_i(t) :=$ {all robots visible from v_i including v_i itself}
$\mathcal{C}_i(t) :=$ smallest enclosing circle of $\mathcal{R}_i(t)$
$c_i(t) :=$ center of $\mathcal{C}_i(t)$
$\{$*keep connectivity*$\}$
$\forall v_j \in \mathcal{R}_i(t) : m_j :=$ the midpoint between $v_i(t)$ and $v_j(t)$
$\forall v_j \in \mathcal{R}_i(t) : \mathcal{D}_j :=$ the circle with radius $\frac{1}{2}$ around m_j, called *limit circle*
seg := the line segment $\overline{v_i(t), c_i(t)}$
$\mathcal{A} := \bigcap_{v_j \in \mathcal{R}} \mathcal{D}_j \cap \text{seg}$
$x :=$ the point in A that minimizes $d(x, c_i(t))$
$\{$*Note that* $\mathcal{A} \neq \emptyset$, *since* $v_i(t) \in \mathcal{A}\}$
$v_i(t+1) := x$

is used to compute a point x on the line segment between $v_i(t)$ and $c_i(t)$ to which v_i finally moves. For each neighbor v_j, v_i computes the midpoint m_j between their positions, and the *limit circle* D_j with center m_j and radius 1/2. As long as both v_i and v_j do not leave this circle, they will be in distance 1 of each other and therefore neighbors at the beginning of the next round. Finally, x is the point on the line segment between $v_i(t)$ and $c_i(t)$ that maximizes the distance that v_i moves under the constraint that v_i does not leave the circle D_j for any neighbor v_j. Since all robots execute this algorithm, this procedure makes sure that two neighboring robots never lose their connection.

Lemma 7.1 (Ando, Suzuki and Yamashita [ASY95, AOSY99]). *If two robots are neighbors in G_t at time t, then they are still neighbors in G_{t+1}. In particular, if G_0 is connected, then G_t is connected for all $t \geq 0$.*

Because of the procedure to keep connectivity, it is possible that a robot does not move far in direction towards its target point. We say that a robot v_j *hinders* another robot v_i from reaching some point p on the line segment between $v_i(t)$ and $c_i(t)$, if v_i would leave D_j when moving to p. If in any round, two robots move to the exact same point, they will stay at a common point for the rest of the execution of the algorithm, because they see the same neighborhood and hence behave exactly the same. Like in Chapter 6, we say that such robots have *fused*.

In [ASY95, AOSY99], the authors have already shown that this algorithm gathers the robots in one point within finite time, but so far no runtime bounds were known. We will now first show a lower bound $\Omega(n^2)$, and then our main result, namely the upper runtime bound of $\mathcal{O}(n^2)$ rounds.

7.2 The Lower Bound

For a lower bound on the number of rounds until gathering when using Go-To-The-Center, consider a configuration with the robots positioned on the boundary of a circle, such that each robot has only two neighbors and the distance between two neighbors on the circle is the same for all robots. In this configuration, all robots have the same local view and so all robots do the same. The robots will therefore still be positioned on the boundary of a circle in the next round. We will use this observation to prove the following result.

Theorem 7.2. *There is a start configuration such that the algorithm takes $\Omega(n^2)$ rounds to gather the robots in one point.*

Proof. Let the robots be positioned on a circle with an initial distance of 1 between two neighboring robots (see Figure 7.1 for an illustration). This means that the initial circumference of the circle is $\approx n$, and its radius is $\approx \frac{n}{2\pi}$. We will show that it takes $\Omega(n^2)$ rounds until the circumference of the circle is reduced to $\frac{2}{3}n$.

If the circumference of the circle is greater than $\frac{2}{3}n$, each robot r has only two neighbors, which are in equal distance d, $\frac{1}{2} < d \leq 1$, from r. The center of the SEC of r's neighborhood is the midpoint between its neighbors. We can therefore compute the distance that r moves as the height h of the equilateral triangle formed by r and its two neighbors. To compute h, let α be the internal angle of the triangle at robot r. Due to the definition of the cosine, $h = \cos(\frac{\alpha}{2}) \cdot d$. In the interval between 0 and $\frac{\pi}{2}$, the cosine can be upper bounded by $\cos(x) \leq -x + \frac{\pi}{2}$. As $0 < \frac{\alpha}{2} < \frac{\pi}{2}$, we can apply this bound and thus $\cos(\frac{\alpha}{2}) \leq -\frac{\alpha}{2} + \frac{\pi}{2}$, resulting in $h \leq \left(-\frac{\alpha}{2} + \frac{\pi}{2}\right) \cdot d$. Moreover, since the robots form a regular polygon with n vertices and the sum of the internal angles of such a polygon is $\pi n - 2\pi$, we get that $\alpha = \pi - \frac{2\pi}{n}$ for all robots. Thus,

$$\begin{aligned}
h &\leq \left(-\frac{\alpha}{2} + \frac{\pi}{2}\right) \cdot d \\
&\leq \left(-\left(\frac{\pi}{2} - \frac{\pi}{n}\right) + \frac{\pi}{2}\right) \cdot d \\
&= \frac{\pi}{n} \cdot d \leq \frac{\pi}{n}
\end{aligned}$$

and the robots move at most a distance of $\frac{\pi}{n}$ in each round. Therefore, it takes at least $\frac{1}{3\pi}n^2$ rounds until the radius is decreased by at least $\frac{1}{3}n$. As the circumference is 2π times the radius of a circle, decreasing the radius by $\frac{1}{3}n$ also decreases the circumference by $\frac{1}{3}n$. Thus, it takes at least $\frac{1}{3\pi}n^2$ rounds until the circumference is decreased to $\frac{2}{3}n$. ■

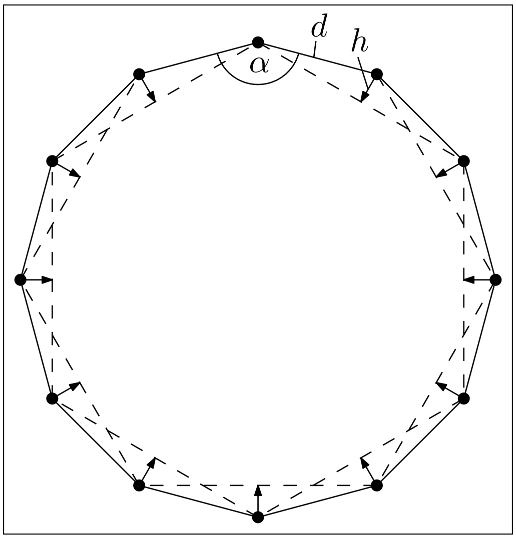


Figure 7.1: A robot configuration on the vertices of a regular convex polygon yields a worst-case running time of the algorithm.

7.3 The Upper Bound

In this section we will show that the robots gather in $\mathcal{O}(n^2)$ rounds. But before we start with the analysis, we state some well-known facts about smallest enclosing circles, on which our analysis will rely heavily.

Proposition 7.3 (Chrystal [Chr85]). *Let $\mathcal{C}$ be the smallest enclosing circle (SEC) of a point set $\mathcal{S}$. Then either*

1. *there are two points $\mathsf{P}, \mathsf{Q} \in \mathcal{S}$ on the circumference of $\mathcal{C}$ such that the line segment $\overline{\mathsf{PQ}}$ is a diameter of $\mathcal{C}$, or*

2. *there are three points $\mathsf{P}, \mathsf{Q}, \mathsf{R} \in \mathcal{S}$ on the circumference of $\mathcal{C}$ such that the center c of $\mathcal{C}$ is inside $\triangle\mathsf{PQR}$, which means that $\triangle\mathsf{PQR}$ is acute-angled.*

Furthermore, the SEC of a set of points is unique.

From this proposition follows directly that the SEC of a point set P is always within the convex hull of P.

The following definition is illustrated in Figure 7.2.

Definition 7.4. *Let $\mathcal{C}$ be the SEC of a set of points $\mathcal{S}$. An arc of $\mathcal{C}$ that contains no points is called a* point-free arc*. The* length *of this arc is defined as the central angle of the arc.*

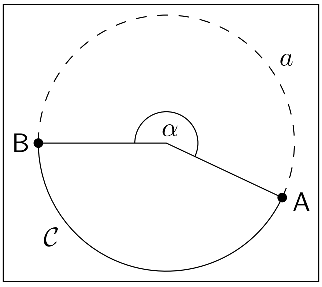


Figure 7.2: The central angle α of an arc a of the circle $\mathcal{C}$ is the angle subtended at the center of $\mathcal{C}$ by the two points A and B delimiting the arc.

Note that the central angle of an arc is greater than π if the arc extends over more than half the circumference of the circle.

Proposition 7.5 (Chrystal [Chr85]). *Let $\mathcal{C}$ be the SEC of a set of $n \geq 2$ points. Then there is no point-free arc with length greater than π.*

With these basics, we can now define how we measure progress. Like for the analysis of MoveInCH, we will use two progress measures.

- As a first progress measure, we will again count the number of rounds in which robots fuse. As we have n robots in the beginning, there can be at most $n-1$ such rounds.

- Since the algorithm is deterministic and it was already proven in Ando, Suzuki and Yamashita's original paper [ASY95] that the robots gather in finite time, we know that, for a given start configuration, the point where the robots gather is fixed. We will call this point the *gathering point* M. We define a circle $\mathcal{N}_t$ with center M and radius R_t for a round t, such that $\mathcal{N}_t$ contains all robots in round t and its radius is minimal. Due to the definition of the algorithm and because the center of the SEC of a point set is always within the convex hull of the point set, the robots never leave the convex hull of their neighbors as well as the global convex hull. R_t can therefore only decrease. We will use R_t as a second progress measure.

  As the robots gather at a point inside the convex hull of the robot positions in any round t, M is inside the convex hull of the robot positions of the start configuration. Moreover, since G_0 is connected, the diameter of the convex hull of the robots in round 0 can be at most $n-1$ and therefore also $R_0 \leq n-1$. The idea of the proof is to show that in a constant number of rounds in which no robots fuse, R_t decreases by at least $\Omega(\frac{1}{n})$.

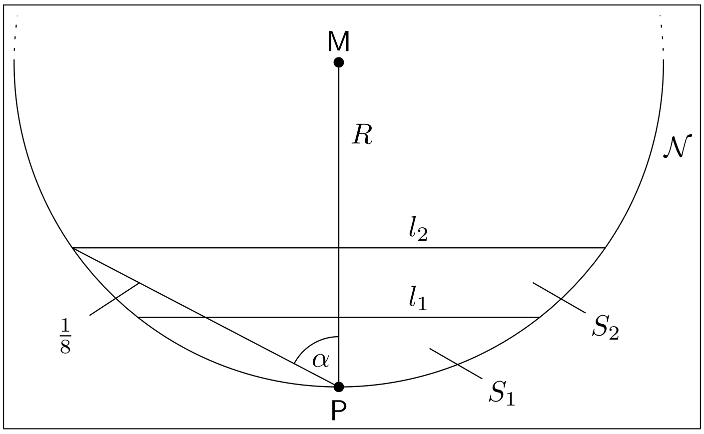


Figure 7.3: The segments S_1 and $S_1 \cup S_2$ of the global SEC are later used to measure the progress of the algorithm.

Using these two progress measures, with $R_0 \leq n-1$ and at most $n-1$ rounds in which robots fuse, it follows directly that the robots gather in $\mathcal{O}(n^2)$ rounds.

From now on, we will consider an arbitrary but fixed round t_0. Let $\mathcal{N} := \mathcal{N}_{t_0}$ and $R := R_{t_0}$. For this round, we introduce some further notions (see Figure 7.3): first, fix an arbitrary point P on the boundary of $\mathcal{N}$ and draw a line between P and M. A line l_2 that is perpendicular to this line defines a circular segment of $\mathcal{N}$. The intersection points of l_2 and the circle $\mathcal{N}$ are in distance $\frac{1}{8}$ from P. Observe that the length of l_2 is upper bounded by $\frac{1}{4}$. We call S_1 the circular segment with half the height of the segment defined by l_2, such that a line l_1 that is parallel to l_2 is its chord. Moreover, we define S_2 to be the area of the segment defined by l_2 minus the area of S_1. The main idea of the analysis is to show that in round t_0 and t_0+1, either two robots fuse or all robots leave S_1. We will conclude that this leads to the desired number of rounds.

The following analysis is divided into geometric prerequisites regarding S_1 and S_2 (Section 7.3.1) and the actual analysis of the algorithm (Section 7.3.2).

7.3.1 Geometric Prerequisites

In this section we want to give prerequisites regarding S_1 and S_2 and smallest enclosing circles with centers in these segments. These will be used later to make a statement about which robots can compute target points inside one of the segments.

Lemma 7.6. *Let x be the length of a chord defining a circular segment S of $\mathcal{N}$. Then any circle $\mathcal{C}$ with its center c in S and radius $r > x$ has an arc outside of $\mathcal{N}$ with a central angle larger than π and thus cannot be the SEC of points only from $\mathcal{N}$.*

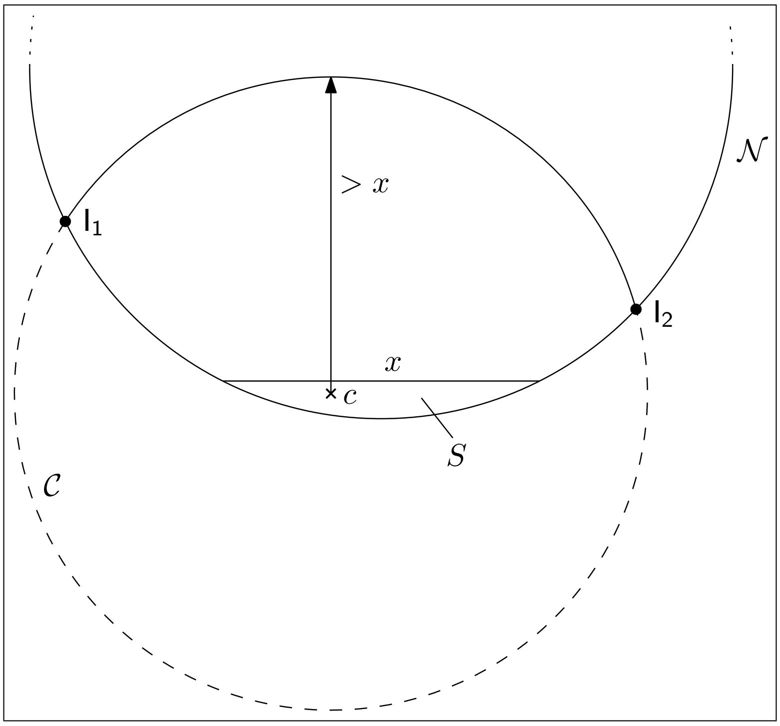


Figure 7.4: A circle with center in S and a radius exceeding the chord length of S intersects with $\mathcal{N}$ outside of S.

Proof. See Figure 7.4 for an illustration of the setting described by the lemma. Since r is larger than the length of the maximum distance between two points in S, both intersection points I_1 and I_2 of the circle $\mathcal{N}$ with any circle with center in S and radius $r > x$ lie outside of S. Because the center c lies in S, it follows that the (longer) arc of $\mathcal{C}$ from I_1 to I_2 outside of $\mathcal{N}$ has a central angle larger than π (the dashed part of the circumference in Figure 7.4). ∎

Since the chord length of $S_1 \cup S_2$ is bounded by $\frac{1}{4}$, the following corollary is immediate.

Corollary 7.7. *The radius of a SEC of a point set $\mathcal{S} \subseteq \mathcal{N}$ with its center in $S_1 \cup S_2$ is at most $\frac{1}{4}$.*

In the following, we will show two geometrical lemmas for the position of the center of a SEC, if the configuration of the underlying points adheres to a few restrictions. The first lemma follows from Corollary 7.7 and will be used to show that if a robot can see a robot that is far away from $S_1 \cup S_2$, it cannot compute a target point inside this circular segment.

Lemma 7.8. *Let $\mathcal{S} \subseteq \mathcal{N}$ be a set of points. Now let* A *be a point in $S_1 \cup S_2$ and* $\mathsf{B} \in \mathcal{S}$ *be a point in distance at least 1 from* A*. Then the center of the SEC of $\mathcal{S}$ cannot lie in the segment $S_1 \cup S_2$.*

Note that A does not need to be in $\mathcal{S}$.

Proof. Assume that the SEC $\mathcal{C}$ has its center c inside $S_1 \cup S_2$. We know from Corollary 7.7 that $\mathcal{C}$ can have at most radius $\frac{1}{4}$. Since the maximum distance of two points in $S_1 \cup S_2$ is bounded by $\frac{1}{4}$, B must have a distance of at least $\frac{3}{4}$ from $S_1 \cup S_2$ in order to be in distance at least 1 from A. Hence, B cannot lie in $\mathcal{C}$. ■

The next lemma is similar to the last one in the sense that it makes a statement about configurations, for which robots cannot compute a target point in S_1. In particular, it will be used for robots that can only see one single robot in $S_1 \cup S_2$. These robots cannot compute a target point in S_1.

Lemma 7.9. *The center of the SEC of a non-empty point set $\mathcal{S} \subseteq \mathcal{N} \setminus (S_1 \cup S_2)$ and a point* $\mathsf{A} \in S_1 \cup S_2$ *cannot lie in the segment S_1.*

Proof. Assume that the SEC $\mathcal{C}$ has its center c inside S_1. We distinguish two cases as given by Proposition 1.

1. $\mathcal{C}$ is defined by two points P_1 and P_2. A must be one of these points, say P_2, otherwise c cannot lie in S_1. Since P_1 cannot lie in S_1 or S_2 by assumption and because the height of S_1 is equal to the height of S_2, the midpoint c of $\overline{\mathsf{AP}_1}$ cannot lie in S_1.

2. $\mathcal{C}$ is defined by three points P_1, P_1 and P_3. A must be one of these points, say P_3, otherwise c cannot lie in S_1. Since $\mathcal{C}$ is the circumcircle of $\triangle\mathsf{P}_1\mathsf{P}_2\mathsf{A}$, it lies on the intersection of the perpendicular bisectors of $\overline{\mathsf{AP}_1}$ and $\overline{\mathsf{AP}_2}$. The centers of these two segments lie outside S_1 and since the perpendicular bisectors intersect in the interior of $\triangle\mathsf{P}_1\mathsf{P}_2\mathsf{A}$ and this triangle is acute, their intersection point also cannot lie in S_1.

This completes the proof. ■

Finally, as the main idea of the analysis is to show that if no robots fuse, S_1 is empty after two rounds, we will need the height of S_1 to compute the progress with respect to R_t within two rounds.

Lemma 7.10. *The segment S_1 has a height h of at least $\frac{1}{128\pi \cdot R} \in \Omega\left(\frac{1}{n}\right)$.*

Proof. We start by computing the angle α (see Figure 7.3 for a definition of α). The circumference of $\mathcal{N}$ is $2\pi R$. Thus, we can position at most $16\pi R$ points on the boundary of $\mathcal{N}$ that are in distance $\frac{1}{8}$ from the points closest to them and that form a regular convex polygon. The internal angle of each of the points of this polygon is equal to 2α. To compute such an internal angle, we use that the sum of the internal angles of a convex polygon is $(m-2)\cdot\pi$, where m is the number of vertices of the polygon. In our case, this is at most $(16\pi R-2)\cdot\pi$. It follows that each angle is at most $\frac{(16\pi R-2)\cdot\pi}{16\pi R} = \pi - \frac{1}{8R}$, and thus $\alpha \leq \frac{\pi}{2} - \frac{1}{16R}$.

Now we can use α and the fact that $\cos(x) \geq -\frac{2}{\pi}x + 1$ in the interval $x \in [0, \frac{\pi}{2}]$ to compute the height h of S_1:

$$\begin{aligned} h = \frac{\cos\alpha}{16} &\geq \frac{\cos\left(\frac{\pi}{2} - \frac{1}{16R}\right)}{16} \\ &\geq \frac{1}{16}\cdot\left(-\frac{2}{\pi}\cdot\left(\frac{\pi}{2} - \frac{1}{16R}\right) + 1\right) \\ &= \frac{1}{128\pi R} \end{aligned}$$

Because $R \leq n$, we have shown $h \in \Omega(\frac{1}{n})$. ■

7.3.2 Gathering Algorithm Analysis

Now we can proceed to the actual analysis of the algorithm. We can use the lemmas from Section 7.3.1 to determine robots that cannot compute a target point in S_1 or $S_1 \cup S_2$. Nevertheless, according to the algorithm, robots do not always reach their target point; it is also possible that they are hindered by other robots. So knowing that a target point is outside S_1 or $S_1 \cup S_2$ does not necessarily mean that the robot actually leaves the respective segment. The following two lemmas show that robots always reach their target point, if it is in $S_1 \cup S_2$, and that they cannot be hindered from leaving S_1 and S_2.

Lemma 7.11. *Robots that compute a target point in $S_1 \cup S_2$ cannot be hindered from reaching it by the limit circle of any other robot.*

Proof. Let v_i be a robot that computes a target point c (which is the center of the SEC $\mathcal{C}$) inside $S_1 \cup S_2$. Then, according to Corollary 7.7, the radius of $\mathcal{C}$ cannot exceed $\frac{1}{4}$ and thus the distance between v_i and c is also upper bounded by $\frac{1}{4}$. Now assume that there is a robot v_e that hinders v_i from reaching c. Since v_e must be a neighbor of v_i, it must also be included in $\mathcal{C}$ and therefore, v_e can have at most distance $\frac{1}{2}$ from v_i. Now let m_e be the midpoint between v_i and v_e and therefore the center of the limit circle that hinders v_i from reaching c. m_e can be at most in distance $\frac{1}{4}$ from v_i. But that means that v_i can move freely in any direction a distance of $\frac{1}{2} - \frac{1}{4} = \frac{1}{4}$ and hence it can reach its target point without being hindered by v_e. ∎

Lemma 7.12. *Robots cannot be hindered from leaving $S_1 \cup S_2$ by the limit circle of any other robot.*

Proof. Let v_i be a robot that computes a target point outside $S_1 \cup S_2$ in round t_0. Now assume for the sake of contradiction that there is one robot v_j that hinders v_i from leaving $S_1 \cup S_2$. This is only possible if v_j is a neighbor of v_i and thus v_j must be within distance 1 of v_i (see the circle $\mathcal{C}_1$ in Figure 7.5 with center v_i and radius 1: v_j must be in $\mathcal{C}_1$). Now let m be the point where v_i would leave $S_1 \cup S_2$ if moving to its target point. According to the algorithm it is only possible that v_i is hindered by v_j to leave $S_1 \cup S_2$, if m is not within distance $\frac{1}{2}$ from the midpoint m_j between v_i and v_j (line 6 – 10 of the algorithm). It follows that m_j cannot be inside the circle $\mathcal{C}_2$ (Figure 7.5) with center m and radius $\frac{1}{2}$. Based on $\mathcal{C}_2$ we can define a circle $\mathcal{C}_3$ which may not contain v_j, if m_j is not in $\mathcal{C}_2$ and m_j is the midpoint between v_i and v_j: $\mathcal{C}_3$'s center is p_i', which is v_i reflected with respect to the point m, and its radius is 1 (see Figure 7.5). Summing up, v_j must be inside of $\mathcal{C}_1$, but outside of $\mathcal{C}_3$. Moreover, the smallest enclosing circle computed by the algorithm has at most radius 1, and so v_i's target point is at most in distance 1 of v_j. It follows that v_i's target point must be on the line between m and p_i', because each point on the straight line through v_i and m beyond p_i' is in distance more than 1 from any point that is in $\mathcal{C}_1$, but not in $\mathcal{C}_3$.

Case 1: v_j is in $S_1 \cup S_2$. Then, because the chord length of $S_1 \cup S_2$ is at most $\frac{1}{4}$, the distance between v_i and v_j is also at most $\frac{1}{4}$. But that means that v_i is at most in distance $\frac{1}{8}$ from the midpoint between v_i and v_j and thus it can move at least distance $\frac{1}{2} - \frac{1}{8} = \frac{3}{8} > \frac{1}{4}$ freely in any direction without being hindered by v_j. But after v_i has moved a distance of $\frac{1}{4}$, it has left $S_1 \cup S_2$ leading to a contradiction.

Case 2: v_j is not in $S_1 \cup S_2$. Since a SEC is defined by two or three points with at least one point on each half of the boundary of the SEC (Proposition 7.5), there must be a robot v_k that is in $S_1 \cup S_2$ and on the boundary of the SEC defining v_i's target point. It follows that v_k can be at most in distance $\frac{1}{4}$ from m. As v_i is also at most in distance $\frac{1}{4}$ from m, so is p_i' and also v_i's target point, which is between m and p_i' (see above). Thus, v_k is at most in distance $\frac{1}{2}$ from v_i's target point. Since v_k is on the boundary of the SEC that defines v_i's target point, it follows that the SEC can have at most a radius of $\frac{1}{2}$. Now, since v_j is outside of $\mathcal{C}_3$ and because the distance between m and p_i' is at most $\frac{1}{4}$ (see above), v_j must be in distance greater than $\frac{1}{2}$ from v_i's target point. Thus, v_j cannot be in the SEC that defines

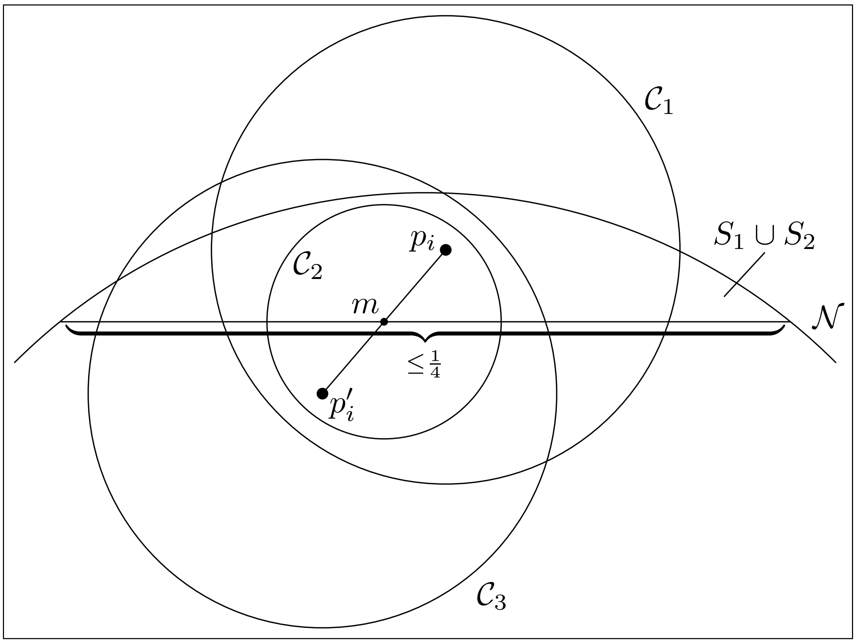


Figure 7.5: Illustration of the proof of Lemma 7.12. The circles indicate where v_j can be positioned: $\mathcal{C}_1$ is a circle with center v_i and radius 1 and must contain v_j. $\mathcal{C}_2$ has center m and radius $\frac{1}{2}$, and $\mathcal{C}_3$'s center is p_i' with radius 1. v_j must not be in $\mathcal{C}_3$.

v_i's target point, which is a contradiction to v_i and v_j being neighbors. It follows that v_j cannot hinder v_i from leaving $S_1 \cup S_2$. ■

With all these prerequisites, we can now show that if no robots fuse, S_1 is empty after two rounds. We first analyze the behavior of some robots in round t_0 in Lemma 7.13, before we plug things together in Lemma 7.14.

Lemma 7.13. *Let $\mathcal{S}$ be a set of robots in round t_0 that are all positioned in or compute a target point in $S_1 \cup S_2$ and that all have a pairwise different neighborhood. Then at most one of those robots is in $S_1 \cup S_2$ at the beginning of the next round.*

Proof. Since all robots from $\mathcal{S}$ have different neighbors, there exists a robot $v_i \in \mathcal{S}$ for which no robot from $\mathcal{S}$ has a set of neighbors that is a subset of the neighbors of v_i. Thus, all robots $v_j \in \mathcal{S} \setminus \{v_i\}$ have a neighbor that is not visible from v_i and therefore in distance more than 1 from v_i. If v_i is positioned in $S_1 \cup S_2$, all robots $v_j \in \mathcal{S} \setminus \{v_i\}$ see a point B in $\mathcal{N}$ (namely the position of the neighbor that v_i cannot see) that is in distance 1 from a point A in $S_1 \cup S_2$ (namely the position of v_i). Lemma 7.8 therefore guarantees that all neighbors of v_i compute a target point outside of $S_1 \cup S_2$. According to Lemma 7.12, no robot is hindered from leaving $S_1 \cup S_2$. Thus, only v_i can stay in $S_1 \cup S_2$.

If v_i is positioned outside $S_1 \cup S_2$, it has its target point in $S_1 \cup S_2$ according to the definition of $\mathcal{S}$. Corollary 7.7 now gives that the radius of v_i's SEC cannot exceed $\frac{1}{4}$ and thus v_i is in distance at most $\frac{1}{4}$ from $S_1 \cup S_2$. Using that the distance between two points in $S_1 \cup S_2$ is at most $\frac{1}{4}$, it follows that all points within $S_1 \cup S_2$ are in distance at most $\frac{1}{2}$ from v_i. Now consider a robot $v_j \in \mathcal{S} \setminus \{v_i\}$ and a neighbor v_k of v_j that is in distance more than 1 from v_i. This robot v_k must then be in distance more than $\frac{1}{2}$ from $S_1 \cup S_2$. Since v_k is v_j's neighbor, we know from Corollary 7.7, that the center of v_j's SEC – its target point – cannot be in $S_1 \cup S_2$ and according to Lemma 7.12 v_j is not hindered from leaving $S_1 \cup S_2$. Since this holds for all robots $v_j \in \mathcal{S} \setminus \{v_i\}$, v_i is the only robot that can be in $S_1 \cup S_2$ in round $t+1$. ■

Lemma 7.14. *If $R_t \geq \frac{1}{2}$, either there are robots that fuse in round t or after two rounds, the segment S_1 does not contain any robots.*

Proof. We consider all robots that are positioned in $S_1 \cup S_2$ or compute a target point in $S_1 \cup S_2$ in round t. We divide this set of robots into two subsets and analyze them separately.

- First, we consider all robots that have a neighbor with the same neighborhood. Thus, for all these robots there is another robot that computes the same target point. Then there are two possibilities: Either one of these target points is in $S_1 \cup S_2$. According to Lemma 7.11, the robots with this target point are not hindered from reaching it and therefore they fuse. If all target points are outside $S_1 \cup S_2$, Lemma 7.12 guarantees that all these robots leave $S_1 \cup S_2$.
- Now consider the robots that have a pairwise different neighborhood. According to Lemma 7.13, at most one of those robots, call it v_i, stays in $S_1 \cup S_2$ during this round.

Thus, if v_i is positioned outside S_1 at the end of round t, we are done. Otherwise, since apart from v_i no robot is still in $S_1 \cup S_2$, we know from Lemma 7.9, that neither v_i nor a neighbor of v_i can compute a target point in S_1 in round $t+1$. Thus, v_i leaves S_1 in round $t+1$ (Lemma 7.12) and none of its neighbors enters S_1. All other robots that are not neighbors of v_i do not see a robot in S_1 and thus they cannot enter S_1. ■

Lemma 7.14 will be used to show that if no robots fuse, R_t decreases by $\Omega\left(\frac{1}{n}\right)$ every two rounds. According to the following Lemma, this procedure stops as soon as $R_t < \frac{1}{2}$.

Lemma 7.15 (Ando, Suzuki and Yamashita [ASY95, AOSY99])**.** *If $R_t < \frac{1}{2}$, the robots have gathered at one point in round $t+1$.*

This lemma holds because if $R_t < \frac{1}{2}$, all robots can see each other and thus all robots compute the same target point. It is shown Ando, Suzuki and Yamashita's original work [ASY95, AOSY99] that the robots do not hinder each other from reaching this point.

Putting everything together, we are now able to prove the final result.

Theorem 7.16. *The robots gather within $\mathcal{O}(n^2)$ rounds.*

Proof. Fix an arbitrary round $t_0 \geq 0$. Since Lemma 7.14 holds for any point on the boundary of N_{t_0}, after two rounds either two robots have fused or all robots must be in distance greater than the height of S_1 from the boundary of N_{t_0}. According to Lemma 7.10, the height of S_1 is at least $\frac{1}{128 \cdot R_t}$ and thus if the robots do not fuse, the radius decreases by at least $\frac{1}{128 \cdot R_t}$, giving that $R_{t+2} \leq R_t - \frac{1}{128 \cdot R_t} \leq R_t - \frac{1}{128 \cdot R_0}$. It follows that after $2 \cdot 128 \cdot (R_0)^2 = 256 \cdot (R_0)^2$ rounds without fusing robots, the radius must be less than $\frac{1}{2}$. Now it takes one round to gather the robots (Lemma 7.15). Moreover, since G_0 is connected, $R_0 \leq n$. There are at most $n-1$ rounds in which robots fuse. The total number of rounds is therefore at most $256 \cdot n^2 + n$. ■

With this runtime bound, we have seen that solving the gathering problem is possible efficiently in a completely local setting and by a very intuitive algorithm. Since each robot is active once and moves at most a distance of 1 in each synchronous round, it follows that the number of neighborhood observations as well as the traveled distance per robot are upper bounded by $\mathcal{O}(n^2)$. In the next chapter, we investigate whether the traveled distance can be reduced if the robots may permanently observe their neighborhood.

Chapter 8

Gathering regarding the traveled distance

We have seen in the previous two chapters that the gathering problem can be solved efficiently in terms of the number of rounds by completely local algorithms. But as for the robot chain problem, we also want to tackle the question how far the robots need to travel for gathering. Therefore, we again let the robots observe their environment continuously and analyze an algorithm which is designed to reduce the traveled distance. This algorithm was introduced in [GWB04] and the authors showed that it gathers the robots in finite time, but no runtime bounds were shown. The algorithm uses the continuous time model as presented in Section 2.2 and used in Chapter 3. It is similar to Move-On-Bisector, and we will therefore call it Gathering-Move-On-Bisector. We will show that the maximum distance traveled when using Gathering-Move-On-Bisector is upper bounded by $\mathcal{O}(min\{OPT^2, n\})$. Like for the robot chain problem, OPT indicates a lower bound for the distance an optimal global algorithm would have to overcome. That is, Gathering-Move-On-Bisector is asymptotically optimal for worst-case instances, for which any global algorithm also needs to cover a distance of $\Omega(n)$. Moreover, for instances which can be solved very quickly by an optimal global algorithm, Gathering-Move-On-Bisector is OPT-competitive.

The analysis of Gathering-Move-On-Bisector is closely related to the analysis of Move-On-Bisector, but it uses also techniques from the analysis of Go-To-The-Center. Therefore, Gathering-Move-On-Bisector is an example for that the techniques which were developed for Move-On-Bisector can be used for further robot formation problems. This raises the hope that they can be transferred to further robot formation problems as well.

Before we start with the analysis and introduce our progress measures, we start with a description of the algorithm.

8.1 The Gathering-Move-On-Bisector Strategy

The GATHERING-MOVE-ON-BISECTOR Strategy [GWB04], which is similar to the MOVE-ON-BISECTOR strategy for the robot chain problem, is as follows. At each point of time, each robot v_i observes all robot positions of those robots within distance 1 (the viewing range) of itself. It then computes the convex hull of these positions ($CH_i(t)$) and performs one of the following actions.

- If v_i is strictly inside this convex hull, it does not move.
- If it is on a line of the border of the convex hull, it moves with this line, keeping the ratio of distances between its two neighbors on the border the same.
- If it is a vertex of $CH_i(t)$, it moves with maximum speed 1 on the angle bisector of the smaller one of the two angles defined by itself and its neighboring two vertices of $CH_i(t)$. We call this angle $\alpha_i(t)$. In the case that v_i has only one neighbor, $\alpha_i(t) = 0$ and v_i moves with maximum speed 1 towards its neighbor.

See Figure 8.1 for an example configuration.

Note that the movement of robots is continuous, but the direction in which a robot moves may change in a non-continuous way, when two robots come into viewing range of each other and one is a new neighbor in the local convex hull of the other. But since the number of changes of the unit disk graph is finite, this only occurs at a finite number of points of time.

8.2 Analysis of the strategy

As already said, we want to analyze the strategy with respect to the maximum traveled distance. Since the robots move with a maximum speed of 1 (and there is at least one robot which always moves with this speed), this distance is equal to the finishing time.

For the analysis of the strategy, we only need one progress measure, which we will call the *length* $\mathfrak{l}(t)$ of a configuration at time t and which is similar to the length of a robot chain. Contrary to the analysis of the robot chain problem, we will not need the height $\mathfrak{h}(t)$ of a configuration as a progress measure. We will nevertheless define the height of a configuration and state the maximum distance traveled in terms of the height instead of the length of a start configuration, using that $\mathfrak{l}(t) \leq \mathfrak{h}(t)^2$

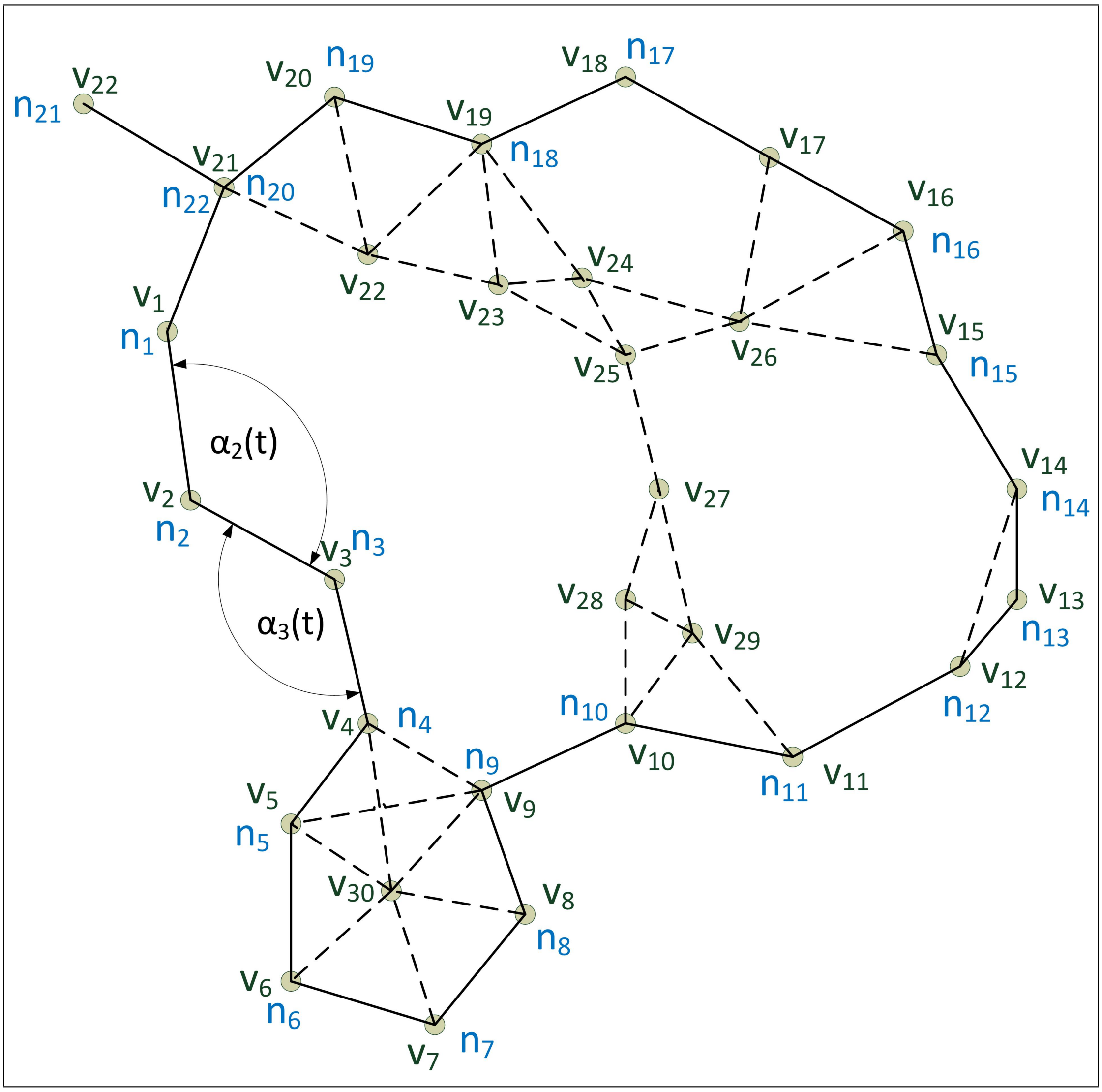


Figure 8.1: Example configuration for Gathering-Move-On-Bisector

(Lemma 8.6), because the height is a more intuitive property of a start configuration than the length. Moreover, through the height we can compare our algorithm with an optimal global algorithm.

For the definition of the height, note that the GATHERING-MOVE-ON-BISECTOR-strategy is deterministic. As used in the analysis of GO-TO-THE-CENTER, it thus exists a gathering point M in which the robots will eventually gather (if gathering is reached, which is still to show).

Definition 8.1 (height). *The* height $\mathfrak{h}(t)$ *of a configuration at time t is the maximum distance between a robot and M at time t.*

Similarly to the MOVE-ON-BISECTOR-strategy, we define the *height-circle* $C_{\mathfrak{h}(t)}$ to be the circle around M with radius $\mathfrak{h}(t)$, which contains all robots. Note that since there is a robot in the start configuration which is in distance $\mathfrak{h}(0)$ from M, and since the gathering point M is inside the convex hull of the robots (the robots never leave the convex hull), even an optimal global algorithm must cover a distance of at least $\mathfrak{h}(0)/2$.

For the definition of the length of a configuration, let G_t again denote the unit disk graph of the robots at time t. G_t has a well-defined outer border, which has the form of a polygon, a line or in parts a polygon and in parts a line (see Fig. 8.1). Nevertheless, we can see the complete outer border as a polygon by using the edges of lines twice. For example, in Fig. 8.1, the vertices of the polygon are formed by the robots on the outer border in the order $v_1, v_2, \ldots, v_{21}, v_{22}, v_{21}, v_1$. The outer border of the unit disk graph is indicated by the solid line, edges of the unit disk graph which are not part of the outer border are indicated by dashed lines. We will call this polygon $P(t)$ from now on.

We will denote the vertices of $P(t)$ by $B_G(t)$ and call them *outer nodes*. Note that one robot can define several vertices and therefore outer nodes, as v_{21} in the example. Moreover, there can be robots on the outer border of the unit disk graph, which are on the line between their neighbors and therefore do not define a vertex of $P(t)$ (for example v_{17}). Finally, if several robots are at the same position, they define only one outer node. Let $m(t) := |B_G(t)|$ be the number of outer nodes at time t, and let $E_B(t) \subseteq B_G(t) \times B_G(t)$ be the set of edges between the outer nodes forming the polygon. We can number the outer nodes from n_1 to $n_{m(t)}$ by starting at one vertex and numbering the nodes counter-clockwise. This way we get that n_i is a neighboring vertex of n_{i+1} in the polygon defined by the outer border. In our example, if starting in the vertex of robot v_1, we get that n_1 is defined by v_1, n_2 by v_2, ..., n_{16} by v_{16}, n_{17} by v_{18}, ..., n_{20} by v_{21}, n_{21} by v_{22} and n_{22} again by v_{21}.

Now we can define the length of a configuration.

Definition 8.2 (length). *The* length $\mathfrak{l}(t)$ *of a configuration at a fixed time* t *is defined as the sum of the distances between neighboring nodes of the polygon* $P(t)$*:* $\mathfrak{l}(t) := d(n_1, n_{m(t)-1}) + \sum_{i=1}^{m(t)} d(n_i(t), n_{i+1}(t))$.

We now want to classify the outer nodes further and define an angle $\gamma_i(t)$ for each outer node, which will be used instead of $\alpha_i(t)$ for bounding the speed with which the length decreases. Consider an outer node n_i which is defined by a robot v_k. According to the algorithm, for each such outer node n_i at time t, exactly one of the following properties is true.

- The robot v_k which defines n_i is a vertex of $CH_{v_k}(t)$. That is, v_k moves on the angle bisector of $\alpha_k(t)$. See for example robot v_2 or v_3 in Figure Fig. 8.1. In this case, we define the angle $\gamma_i(t)$ for the outer node n_i as $\gamma_i(t) := \alpha_k(t) < \pi$.

- v_k is not a vertex of $CH_k(t)$ (for example robot v_4 or v_{21}) and thus it does not move. Here we define $\gamma_i(t) := \pi$. Note that for robot v_{21}, this means that the angles γ of both outer nodes defined by it ($\gamma_{21}(t)$ and $\gamma_{23}(t)$) are set to π.

Note that it is not possible that an outer node is on the straight line between its neighbors on the border of its local convex hull, since we have excluded that such robots define an outer node.

Now we can show that the length decreases depending on the $\gamma_i(t)$'s.

Lemma 8.3. *Given two neighboring outer nodes* n_i *and* n_j *at an arbitrary time* t_0*, in which the unit disk graph does not change. Then the distance between* n_i *and* n_j *decreases with speed at least* $\cos\frac{\gamma_i(t_0)}{2} + \cos\frac{\gamma_j(t_0)}{2} \geq 0$.

Proof. The proof is similar to the one for Move-On-Bisector in Section 3.1. We define $D : \mathbb{R}_{\geq 0} \to \mathbb{R}^2, t \mapsto n_j - n_i$ and $\mathsf{d} : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}, t \mapsto ||D(t)||$. That is, $D(t)$ is the vector from n_i to n_j and $\mathsf{d}(t)$ the distance between n_i and n_j at time t. We want to show that $\mathsf{d}'(t_0) = -\left(\cos\frac{\gamma_i(t_0)}{2} + \cos\frac{\gamma_j(t_0)}{2}\right)$ for an arbitrary but fixed point of time t_0. We will refer to the x- and y-component of $D(t) \in \mathbb{R}^2$ in the following by $D_x(t)$ and $D_y(t)$ respectively. Similarly, we denote the x-coordinate of the position of outer node n_i at time t by $n_{i,x}(t)$ and its derivation by $n'_{i,x}(t)$.

By translating and rotating the coordinate system, we can w.l.o.g. assume $n_i(t_0) = (0,0)$ and $n_j(t_0) = (\mathsf{d}(t_0), 0)$. As the proof is similar to the one for Move-On-Bisector in Section 3.1, Fig. 3.1 can be used as an illustration for the following computations.

First consider n_i, which is defined by robot v_k. We will show that $n'_{i,x}(t_0) = \cos\frac{\gamma_i(t_0)}{2}$. Due to the classification of the outer nodes, we can distinguish the following cases:

1. The robot v_k which defines n_i is a vertex of $CH_k(t)$. Then v_k and therefore also n_i moves in direction of the angle bisector of $\alpha_k(t_0)$, and the velocity vector of n_i at time t_0 is given by

$$n_i'(t_0) = \left(+\cos\frac{\alpha_k(t_0)}{2}, \pm\sin\frac{\alpha_k(t_0)}{2}\right) = \left(+\cos\frac{\gamma_i(t_0)}{2}, \pm\sin\frac{\gamma_i(t_0)}{2}\right).$$

2. v_k is not a vertex of $CH_k(t)$. Then v_k does not move, and the movement vector of n_i is $n_i'(t_0) = (0,0)$. Thus, $n_{i,x}'(t_0) = 0 = \cos\frac{\gamma_j(t_0)}{2}$.

Analogously it can be shown that $n_{j,x}'(t_0) = -\cos\frac{\gamma_j(t_0)}{2}$.

Basic analysis now gives us the following equation for the first derivation of d at a time $t \in \mathbb{R}_{\geq 0}$[1]

$$\mathsf{d}'(t) = \begin{pmatrix} \frac{D_x(t)}{\mathsf{d}(t)} & \frac{D_y(t)}{\mathsf{d}(t)} \end{pmatrix} \cdot \begin{pmatrix} D_x'(t) \\ D_y'(t) \end{pmatrix}$$

Using that we have $D_y(t_0) = 0$ and $D_x(t_0) = \mathsf{d}(t_0)$ we finally get

$$\begin{aligned} \mathsf{d}'(t_0) = D_x'(t_0) = & \quad (v_j - v_i)_x'(t_0) = \left(n_{j,x}'(t_0) - n_{i,x}'(t_0)\right) \\ = & \quad -\left(\cos\tfrac{\gamma_i(t_0)}{2} + \cos\tfrac{\gamma_j(t_0)}{2}\right). \end{aligned}$$

Therefore, the distance between n_i and n_j reduces at time t with speed at least $\cos(\frac{\gamma_i(t_0)}{2}) + \cos(\frac{\gamma_j(t_0)}{2})$. Furthermore, since we have $\gamma_i(t) \in [0,\pi]$ for any $t \in \mathbb{R}_{\geq 0}$ and $i \in \{1,\dots,m(t)\}$, this speed is not negative. ∎

Thus, as for the MOVE-ON-BISECTOR strategy for the robot chain problem, we can give the speed with which the length decreases as the sum over all edges of the polygon: the length decreases with speed $\sum_{i=1}^{m(t)} \cos(\frac{\gamma_i(t)}{2}) + \cos(\frac{\gamma_{i+1}(t)}{2})$. But contrary to the chain, we can bound the sum of the angles $\gamma_i(t)$, since the outer nodes forms a polygon. If $m(t)$ is the number of vertices of the polygon $P(t)$, the sum of its inner angles is $(m(t)-2)\pi$. Now $\gamma_i(t)$ is not always an inner angle of n_i of the polygon $P(t)$. Let us see when this can happen. If n_i forms a *convex* angle of $P(t)$, the inner angle of $P(t)$ at n_i is $\alpha_i(t) = \gamma_i(t)$. If n_i forms a *reflex* angle of $P(t)$, the inner angle of $P(t)$ at n_i is at least $\pi > \gamma_i(t)$. The same is true if n_i is no vertex of its local convex hull at all: The inner angle of $P(t)$ is at least π, which is greater than or equal to $\gamma_i(t) = \pi$. Thus, we know that $\sum_{i=1}^{m(t)} \gamma_i \leq (m-2)\pi$.

Lemma 8.4. *At an arbitrary time t, in which the unit disk graph does not change, $\mathsf{l}(t)$ decreases with speed* 4.

[1] Note that $\mathsf{d}(t) \neq 0$, since several robots at one position define only one outer node.

Proof. Each edge (n_i, n_{i+1}) of the polygon $P(t)$ decreases with speed $\cos(\frac{\gamma_i}{2}) + \cos(\frac{\gamma_{i+1}}{2})$ (Lemma 8.3). Using that $\cos(x) \geq -2/\pi \cdot x + 1$ for $0 \leq x \leq \pi/2$, the length decreases with total speed

$$\begin{aligned}
\sum_{i=1}^{m(t)} 2\cos(\frac{\gamma_i}{2}) &\geq 2\sum_{i=1}^{m(t)} -\frac{2}{\pi}\frac{\gamma_i}{2} + 1 \\
&= 2m(t) - \frac{2}{\pi}\sum_{i=1}^{m(t)} \gamma_i \\
&\geq 2m(t) - \frac{2}{\pi}\pi(m(t) - 2) \\
&= 4.
\end{aligned}$$

■

The Lemmas 8.3 and 8.4 show that the length decreases with speed 4 as long as the unit disk graph does not change. The next lemma shows that at these (finite) points in time, the length does not increase.

Lemma 8.5. *The length $\mathfrak{l}(t)$ decreases monotonically.*

Proof. We have already seen in Lemma 8.3, that the length of the chain decreases at all times when the unit disk graph does not change. So let us now see how the border of the unit disk graph and therefore the polygon $P(t)$ can change topologically and what influence this has on the length. For this, note first that due to the definition of the algorithm, a robot which does not define an outer node in the beginning may move on a line of the border, which does not change the length, but it can never become a vertex of the border of the unit disk graph and therefore define an outer node. That is, there are only two possibilities how the border of the unit disk graph can change topologically:

- Robots can move in a way that they stop to define an outer node: when a robot v_k, which defines an outer node n_i, reaches the line between its two neighboring outer nodes. In this case, the distance between its neighbors n_{i-1} and n_{i+1} (which will then be n_{i-1} and n_i) is set to $d(n_{i-1}, n_i) + d(n_i, n_{i+1})$. Thus, the length does not change.

- Edges of the unit disk graph can never be deleted. But edges on the border can be replaced by another edge, when two robots on the border come into viewing range of each other (e.g. n_8 and n_{11} in Fig. 8.1). The robots in between (v_9, v_{10}) stop to define outer nodes, and the length $\mathfrak{l}(t)$ is decreased.

Thus, in both cases, the length is not increased. ■

Finally, before we can bound the time it takes for gathering, we need to have a relationship between $\mathfrak{l}(t)$ and $\mathfrak{h}(t)$ (Lemma 8.6) and we show that a robot can define only a constant number of outer nodes (Lemma 8.7).

Lemma 8.6. $\mathfrak{l}(t) \in \mathcal{O}(\mathfrak{h}(t)^2)$.

Proof. Consider the polygon $P(t)$. Since it is formed by the outer border of the unit disk graph of the robots, no robot can see any other robot outside $P(t)$. Thus, we can define an area $A(t)$ around $P(t)$, which contains all points outside $P(t)$ in distance at most $\frac{1}{4}$ from $P(t)$. This area cannot contain any robot, since it would be seen by at least one robot on the border of the unit disk graph. See Fig. 8.2 as an example. Moreover, $A(t)$ never intersects with itself, and thus for a length $\mathfrak{l}(t)$, $A(t)$ defines an area of size at least $\frac{1}{4}\mathfrak{l}(t)$ which does not contain any robots.

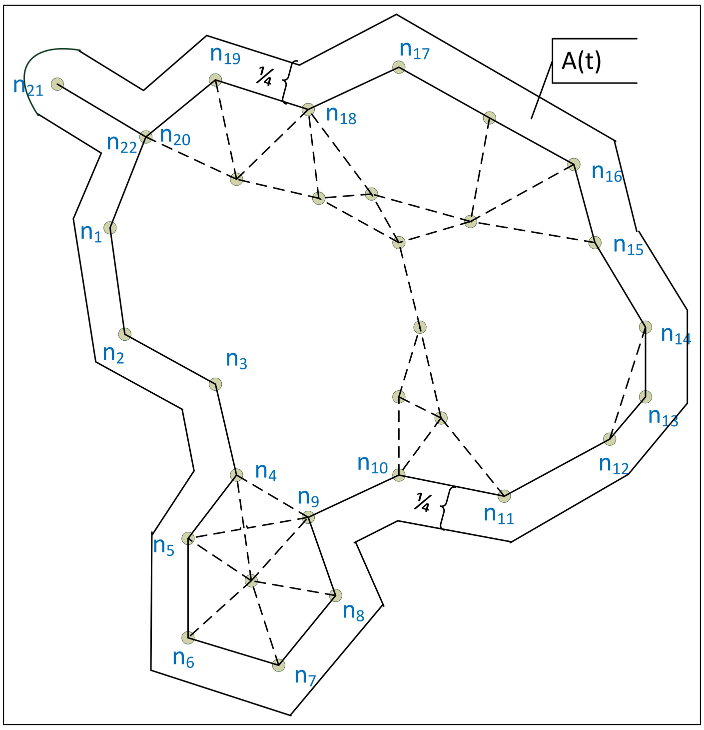


Figure 8.2: The area $A(t)$ is indicated for the example configuration

Now since the robots are all contained in the height-circle $C_{\mathfrak{h}(t)}$ (the circle around M with radius $\mathfrak{h}(t)$), the area of $A(t)$ is completely inside the circle $C_{\mathfrak{h}(t)+\frac{1}{4}}$ around M with radius $\mathfrak{h}(t) + 1/4$. This implies that the area of this circle must be at least the area of $A(t)$, which is at least $\frac{1}{4}\mathfrak{l}(t)$. Thus, we get

$$\frac{1}{4}\mathfrak{l}(t) \leq \pi \left(\mathfrak{h}(t) + \frac{1}{4}\right)^2 = \pi \left(\mathfrak{h}(t)^2 + \frac{1}{2}\mathfrak{h}(t) + \frac{1}{16}\right)$$

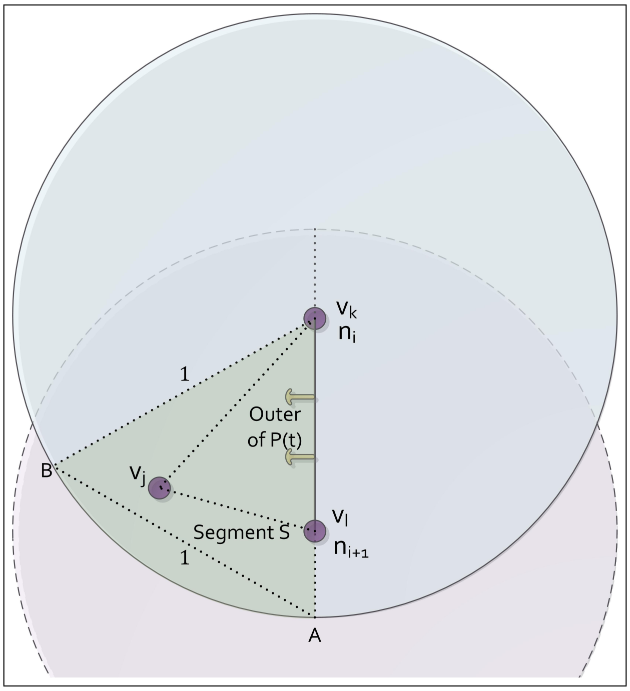


Figure 8.3: Illustration to the proof of Lemma 8.7

and $\mathfrak{l}(t) \in \mathcal{O}(\mathfrak{h}(t)^2)$. ■

Lemma 8.7. *Each robot v_k can define at most 6 outer nodes.*

Proof. First note that if v_k defines x outer nodes, each of the outer nodes has a left and a right neighbor on the border of the polygon. These neighbors are either outer nodes themselves, or they are robots which are on the straight line between two outer nodes. Moreover, the x right neighbors of the x outer nodes are all distinct.

Now let n_i be an outer node defined by v_k, and n_{i+1} its right neighbor (w.l.o.g. let it also be an outer node, this simplifies the notation) which is defined by v_l. Let the the outer of the polygon lie in clockwise direction of v_l when looking from v_k (see Fig. 8.3).

We will show that no robot can be in clockwise direction from v_l which is visible by both v_k and v_l. For the sake of contradiction, assume that there is such a robot v_j (see Fig. 8.3). Then the unit disk graph contains the edges (v_k, v_j) and (v_j, v_l). But then the edge (v_k, v_l) cannot define the outer border of $P(t)$ in clockwise direction.

So we have found an area which cannot contain any robot. Consider the sector S of the visibility circle of v_k which is defined by two points A and B, where A is the point closest to v_l on the circle defining the visibility range of v_k, and B is the point on the circle defining v_k's visibility range in distance 1 in clockwise direction from A (see again Fig. 8.3).This sector is completely contained in the area which cannot contain any robot. The inner angle of the sector at v_k is $\pi/3$. Thus, v_l defines a sector with inner angle $\pi/3$ which cannot contain any robots.

Now consider another outer node defined by v_k, and its right neighbor. Call it v_s. Then the sectors defined by v_s and v_l cannot intersect, since otherwise v_s would be in v_l's sector or the other way round. Thus, all right neighbors of outer nodes defined by v_k define a sector of the visibility range of v_k which are mutually distinct. Since the inner angle of these sectors is $\pi/3$, there can be at most 6 such sectors and therefore also at most 6 right neighbors. It follows that v_k can define at most 6 outer nodes. ■

Now we have all prerequisites for our runtime bound.

Theorem 8.8. *After time $\mathcal{O}(l) \subseteq \mathcal{O}(\min\{OPT^2, n\})$, the robots have gathered in one point.*

Proof. Except for a finite number of points in time, which in sum take time 0 and in which the length does not incease (Lemma 8.5), the length decreases with speed 4 (Lemma 8.4). That is, after time $1/4 \cdot l$ the robots have gathered. For the upper bound of n, we use that each robot can define at most 6 outer nodes (Lemma 8.7). Thus, the polygon $P(0)$ can have at most $6n$ edges, each with a length of at most 1 (if there is an edge which is longer than 1, a robot must be on this line). That is, $\mathfrak{l}(t) \leq 6n$, resulting in an upper bound of $\mathcal{O}(n)$. For the upper bound of OPT^2, we use that $\mathfrak{l}(t) \in \mathcal{O}(\mathfrak{h}(t)^2)$ (Lemma 8.6) and $OPT \geq \mathfrak{h}(t)/2$. The theorem follows. ■

Summing up, the GATHERING-MOVE-ON-BISECTOR strategy is very efficient in terms of the maximum traveled distance. For worst case instances, for example robots positioned on a line, it is optimal (a traveled distance of $\mathcal{O}(n)$ is needed by GATHERING-MOVE-ON-BISECTOR as well as by any optimal global algorithm), and for instances which can be solved with a shorter traveled distance by an optimal global algorithm, it is OPT-competitive. This result combined with the runtime of GO-TO-THE-CENTER raises the hope that there is also an algorithm which is efficient in terms of both quality measures.

Technically, the GATHERING-MOVE-ON-BISECTOR strategy shows a strength of the MOVE-ON-BISECTOR strategy: the techniques used for its analysis can be adapted to further robot formation problems.

Chapter 9

Conclusion and Outlook

The goal of this thesis was to investigate how efficient robot formation problems can be solved by very simple robots with a local view. The question behind was how far the local view restricts the coordination of the robots. While the considered problems are easy to solve for robots with a global view and the same abilities as used in this thesis, one single robot with a local view sometimes cannot even decide whether the desired formation has been achieved. The challenge therefore lies in the coordination of robots with very limited information, such that globally the formation is nevertheless achieved. Moreover, it should be achieved efficiently.

We have investigated two robot formation problems with respect to two quality measures of such robotic systems: the number of rounds and the movement distance. The formation problems were chosen in such a way that they capture two completely different aspects. While the gathering problem is about the simplest formation one can think of, a point, it captures the difficulty of changing neighborhoods. On the other hand, the robot chain problem uses fixed neighborhoods, but its goal is to achieve the non-trivial formation of a line.

In order to analyze the number of rounds, we used existing time and movement models based on the Look-Compute-Move model, which are commonly used in the literature. On the other hand, a continuous movement model was used to investigate the movement distance. Although it is close to some practical applications, this model is uncommon for robot formation problems. To the best of our knowledge, we are the first to analyze the movement distance and finishing time of algorithms for robot formation problems in this model. The techniques used for the analysis of Move-On-Bisector seem to be transferable to further formation problems. We have shown in this thesis that this is possible for gathering. Finally, by restricting the movement distance per time step in the discrete Look-Compute-Move model, we achieved a possibility to investigate the spectrum between the classical discrete model and the continuous model. This way we were able to design strategies which are efficient in terms of both quality measures.

In the literature, the efficiency of robot formation problems is normally analyzed with respect to worst case instances, such that the considered algorithm performs bad under the used quality measure (which is mostly the number of rounds). Like this, upper bounds for the efficiency of the algorithm are achieved. We also did this, but for the movement distance we could go even further: we compared the quality of the algorithm for each fixed instance with that of an optimal global algorithm. We could give competitive factors for our algorithm, like it is usual for approximation and online algorithms.

Several ideas for future work can be derived directly from this conclusion. We were able to give competitive factors for the movement distance in the continuous model, but so far, to the best of our knowledge, no competitive factors are known for further robot formation problems or with respect to the number of rounds. But the analysis of competitiveness would be of great interest: it captures explicitly what prize we have to pay for a local view. Competitive factors for further robot formation problems and for all types of quality measures would therefore be of high interest.

We have analyzed the movement distance in the continuous time and movement model for the robot chain problem, and shown that the techniques can be used for the gathering problem as well. But the movement distance is also an important quality measure for further robot formation problems. Can the techniques directly be applied to further problems, or do new techniques have to be developed? And are there algorithms that are better than MOVE-ON-BISECTOR? Moreover, it remains open so far whether our bounds on the movement distance of $\mathcal{O}(\min\{n, (OPT+d)\log n\})$ for the robot chain problem and $\mathcal{O}(\min\{n, OPT^2\})$ for the gathering problem can be improved.

Similarly, we have combined both quality measures for the robot chain problem, but this is still open for the gathering problem. Is there an algorithm which preserves a good number of rounds, but reduces the traveled distance? In particular, is there an algorithm which needs $O(n^2)$ number of rounds for gathering and a traveled distance only linear in the number of robots? And if not, what trade-off between the two quality measures can be found? Similar questions can be asked for further robot formation problems.

Concerning the gathering problem, it is an open question whether there exist local algorithms which need fewer rounds. Are there strategies which need considerably less than a quadratic number of rounds? The Hopper-strategy for the robot chain problem ([KM09]) indicates that it might help if the robots leave the local convex hull of their neighbors.

On the other hand there are no nontrivial lower bounds known for classes of local algorithms for gathering, the robot chain problem or other formation problems. One would need a clean definition of asynchronous or synchronous local strategies. A crucial property restricting such strategies is that connectivity has to be maintained. Just looking at the start configuration of the lower bound instance from Section 7.2 for the gathering problem, for example, and only demanding connectivity for this specific start configuration is not sufficient: consider the synchronous algorithm in which each point moves in the direction of the target point of our algorithm, but goes beyond this point until the distance to its neighbors is 1. This algorithm maintains connectivity for our specific start configuration, but needs only a linear number of rounds, if the start configuration positions neighboring robots in distance $\frac{2}{3}$ on the cycle. Similar results can be shown for asynchronous strategies with specific activation policies. Such examples demonstrate that the connectivity constraint has to be reflected much more severely in lower-bound models for local strategies.

Generally, it is an interesting problem to investigate to which extend our strategies or further strategies are robust under inaccuracies of sensors and actors. For example, the positions of neighboring robots normally cannot be determined accurately. Similarly, robots normally do not have a fixed viewing range, but this range can vary depending on the surrounding conditions. How to formally model such inaccuracies? For which input instances are the strategies robust? Do we have to modify existing strategies? We certainly have to assume that input instances have the property that neighboring robots have distance at most $1 - \gamma$, where $\gamma \in (0, 1)$ is chosen dependent on parameters describing the accuracy.

Finally, the dynamics which occur in this thesis can all be controlled by our algorithms. But what happens if we add external dynamics to the problem, which cannot be controlled by us? Under which circumstances is it still possible to maintain the connectivity, and which goals can be achieved? For example, consider four externally controlled robots forming a square, with our mobile robots being positioned on the points of a grid inside the square. If the four externally controlled robots move according to the same trajectory, such that the form of the square is maintained, can our mobile robots stay inside the square and keep the connectivity? And what if the form of the square is changed to a rectangle with the same area? A lot of different scenarios like this can be imagined when adding external dynamics.

Bibliography

[AOSY99] Hideki Ando, Yoshinobu Oasa, Ichiro Suzuki, and Masafumi Yamashita. Distributed memoryless point convergence algorithm for mobile robots with limited visibility. *IEEE Transactions on Robotics and Automation*, 15(5):818–828, 1999.

[AP04] 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.

[ASY95] Hideki Ando, Yoshinobu Suzuki, and Masafumi Yamashita. Formation and agreement problems for synchronous mobile robots with limited visibility. In *Proceedings of the 1995 IEEE International Symposium on Intelligent Control, ISIC 1995*, pages 453–460, August 1995.

[BDKM11a] Philipp Brandes, Bastian Degener, Barbara Kempkes, and Friedhelm Meyer auf der Heide. Energy-efficient strategies for building short chains of mobile robots locally. In *SIROCCO '11: Proceedings of the 18th International Colloquium on Structural Information and Communication Complexity*, pages 138–149, 2011.

[BDKM11b] Philipp Brandes, Bastian Degener, Barbara Kempkes, and Friedhelm Meyer auf der Heide. Energy-efficient strategies for building short chains of mobile robots locally. *Invited for submission to a Special Issue of Theoretical Computer Science*, 2011.

[CFPS03] Mark Cieliebak, Paola Flocchini, Giuseppe Prencipe, and Nicola Santoro. Solving the Robots Gathering Problem. In *Automata, Languages* *and Programming: 30th International Colloquium, ICALP 2003*, pages 1181–1196, 2003.

[CGP09] Jurek Czyzowicz, Leszek Gasieniec, and Andrzej Pelc. Gathering few fat mobile robots in the plane. *Theoretical Computer Science*, 410(6-7):481–499, 2009.

[Cha09] Bernard Chazelle. Natural algorithms. In *Proceedings of the Twentieth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2009*, pages 422–431, 2009.

[Chr85] George Chrystal. On the problem to construct the minimum circle enclosing n given points in a plane. In *Proceedings of the Edinburgh Mathematical Society, Third Meeting*, pages 30–35, 1885.

[CLDF+11a] Andreas Cord-Landwehr, Bastian Degener, Matthias Fischer, Martina Hüllmann, Barbara Kempkes, Alexander Klaas, Peter Kling, Sven Kurras, Marcus Märtens, Friedhelm Meyer auf der Heide, Christoph Raupach, Kamil Swierkot, Daniel Warner, Christoph Weddemann, and Daniel Wonisch. A New Approach for Analyzing Convergence Algorithms for Mobile Robots. In *Automata, Languages and Programming - 38th International Colloquium, ICALP 2011, Proceedings, Part II*, pages 650–661, 2011.

[CLDF+11b] Andreas Cord-Landwehr, Bastian Degener, Matthias Fischer, Martina Hüllmann, Barbara Kempkes, Alexander Klaas, Peter Kling, Sven Kurras, Marcus Märtens, Friedhelm Meyer auf der Heide, Christoph Raupach, Kamil Swierkot, Daniel Warner, Christoph Weddemann, and Daniel Wonisch. Collisionless Gathering of Robots with an Extent. In *SOFSEM 2011: Theory and Practice of Computer Science - 37th Conference on Current Trends in Theory and Practice of Computer Science*, pages 178–189, 2011.

[CMN04] Ioannis Chatzigiannakis, Michael Markou, and Sotiris Nikoletseas. Distributed circle formation for anonymous oblivious robots. In *Experimental and Efficient Algorithms, Third International Workshop, WEA 2004*, pages 159–174, 2004.

[CP04] Reuven Cohen and David Peleg. Robot Convergence via Center-of-Gravity Algorithms. In *SIROCCO '04: Proceedings of the 11th International Colloquium on Structural Information and Communication*

Complexity, volume 3104 of *Lecture Notes in Computer Science*, pages 79–88. Springer Berlin / Heidelberg, 2004.

[CP05] Reuven Cohen and David Peleg. Convergence Properties of the Gravitational Algorithm in Asynchronous Robot Systems. *SIAM Journal on Computing*, 34(6):1516–1528, 2005.

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

[DK02] Xavier Défago and Akihiko Konagaya. Circle formation for oblivious anonymous mobile robots with no common sense of orientation. In *Proceedings of the 2002 Workshop on Principles of Mobile Computing, POMC 2002*, pages 97–104, 2002.

[DKKM10] 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 *SIROCCO '10: Proceedings of the 17th International Colloquium on Structural Information and Communication Complexity*, pages 168–182, 2010.

[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: Proceedings of the 23rd annual ACM symposium on parallel algorithms and architectures*, pages 139–147, 2011.

[DKLM06] Miroslaw Dynia, Jaroslaw Kutylowski, Pawel 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 (BICC '06)*, pages 137–146, 2006.

[DKM10] Bastian Degener, Barbara Kempkes, and Friedhelm Meyer auf der Heide. A local O(nˆ2) gathering algorithm. In *SPAA '10: Proceedings of the 22nd ACM symposium on parallelism in algorithms and architectures*, pages 217–223, 2010.

[DKMS07] Miroslaw Dynia, Jaroslaw Kutylowski, Friedhelm Meyer auf der Heide, and Jonas Schrieb. Local Strategies for Maintaining a Chain of Relay Stations between an Explorer and a Base Station. In *SPAA '07: Proceedings of the 19th annual ACM symposium on Parallel algorithms and architectures*, pages 260–269, New York, USA, 2007. ACM Press.

[DP09] Yoann Dieudonné and Franck Petit. Self-stabilizing Deterministic Gathering. In *Algorithmic Aspects of Wireless Sensor Networks, 5th International Workshop, ALGOSENSORS 2009*, pages 230–241, 2009.

[FPSW05] 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.

[GWB04] Noam Gordon, Israel A. Wagner, and Alfred M. Bruckstein. Gathering Multiple Robotic A(ge)nts with Limited Sensing Capabilities. In *Ant Colony Optimization and Swarm Intelligence, 4th International Workshop, ANTS 2004*, pages 142–153, 2004.

[IIKO09] Taisuke Izumi, Tomoko Izumi, Sayaka Kamei, and Fukuhito Ooshita. Randomized Gathering of Mobile Robots with Local-Multiplicity Detection. In *Proceedings of the 11th International Symposium on Stabilization, Safety, and Security of Distributed Systems (SSS 2009)*, pages 384–398, 2009.

[IKIW07] Taisuke Izumi, Yoshiaki Katayama, Nobuhiro Inuzuka, and Koichi Wada. Gathering Autonomous Mobile Robots with Dynamic Compasses: An Optimal Result. In *Distributed Computing, 21st International Symposium, DISC 2007*, pages 298–312, 2007.

[JLM03] Ali Jadbabaie, Jie Lin, and A. Stephen Morse. Coordination of groups of mobile autonomous agents using nearest neighbor rules. *IEEE Transactions on Automatic Control*, 48:988–1001, 2003.

[Kat11] Branislav Katreniak. Convergence with Limited Visibility by Asynchronous Mobile Robots. In Adrian Kosowski and Masafumi Yamashita, editors, *SIROCCO '11: Proceedings of the 18th International Colloquium on Structural Information and Communication Complexity*, pages 125–137. LNCS 6796, 2011.

[KM09] Jaroslaw 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.

[KM11] Peter Kling and Friedhelm Meyer auf der Heide. Convergence of Local Communication Chain Strategies via Linear Transformations. In *SPAA '11: Proceedings of the 23rd ACM symposium on parallelism in algorithms and architectures*, pages 159–166, 2011.

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

[Kut07] Jaroslaw Kutylowski. *Using Mobile Relays for Ensuring Connectivity in Sparse Networks*. Dissertation, International Graduate School of Dynamic Intelligent Systems, 2007.

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

[MS08] Friedhelm Meyer auf der Heide and Barbara Schneider. Local Strategies for connecting stations by small robotic networks. In *IFIP 20th World Computer Congress, TC10: 2nd IFIP International Conference on Biologically Inspired Computing (BICC '08)*, pages 95–104, 2008.

[NPGF04] Hoa G. Nguyen, Narek Pezeshkian, Anoop Gupta, and Nathan Farrington. Maintaining Communication Link for a Robot Operating in a Hazardous Environment. In *Proceedings of the 10th International Conference on Robotics and Remote Systems for Hazardous Environments*. American Nuclear Society, 2004.

[NPR+03] Hoa G. Nguyen, Narek Pezeshkian, Scott M. Raymond, Anoop Gupta, and Jonathan Michael Spector. Autonomous communication relays for tactical robots. In *Proceedings of ICAR 2003: The International Conference on Advanced Robotics*, pages 35–40, 2003.

[PNB07] Narek Pezeshkian, Hoa G. Nguyen, and Aaron Burmeister. Unmanned ground vehicle radio relay deployment system for non-line-of-sight operations. In *RA '07 Proceedings of the 13th IASTED International Conference on Robotics and Applications*, pages 501–506, 2007.

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

[Rey87] Craig W. Reynolds. Flocks, Herds, and Schools: A Distributed Behavioral Model. In *Computer Graphics*, volume 21 (4), pages 25–34, 1987.

[SDY06] Samia Souissi, Xavier Défago, and Masafumi Yamashita. Gathering Asynchronous Mobile Robots with Inaccurate Compasses. In *Principles of Distributed Systems, 10th International Conference, OPODIS 2006*, pages 333–349, 2006.

[STM10] Hisayoshi. Sugiyama, Tetsuo Tsujioka, and Masashi Murata. Integrated operations of multi-robot rescue system with ad hoc networking. In *IEEE International Workshop on Safety Security and Rescue Robotics (SSRR '10)*, pages 1–6, 2010.

[SWC05] Hong Shi, Long Wang, and Tianguang Chu. Coordination of multiple dynamic agents with asymmetric interactions. In *Proceedings of the International Symposium on Intelligent Control*, pages 1423 – 1428, 2005.

[SY93] Ichiro Suzuki and Masafumi Yamashita. Formation and agreement problems for anonymous mobile robots. In *Proceedings of the 31st Annual Allerton Conference on Communication, Control, and Computing*, pages 93–102, September 1993.

[SY99] Ichiro Suzuki and Masafumi Yamashita. Distributed Anonymous Mobile Robots: Formation of Geometric Patterns. *SIAM Journal on Computing*, 28(4):1347–1363, 1999.