A Tight Runtime Bound for Synchronous Gathering of Autonomous Robots with Limited Visibility

Ζυγομήτρος Ευάγγελος
ΜΠΛΑ 201118

Ιούνιος, 2013

The Problem (1)

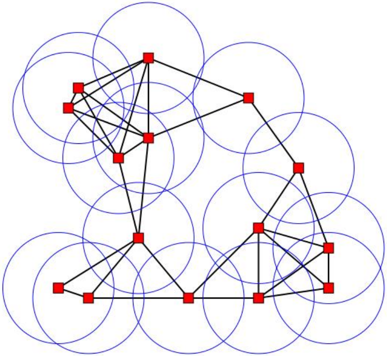
Gathering n autonomous robots, modeled as points in the 2-dimensional Euclidean plane, at a single point.
We are interested in the concurrent version of the problem:

- in each synchronous round, every robot observes the plane and the other robots, decides where to move, and moves there, concurrently with all the other robots.
- the next round does not start before the last movement has finished.

We study the distributed version of the problem, where each robot has a limited viewing range and can only observe other robots that are within unit distance of its position (if robots have full visibility, the problem is trivial as all robots can compute the unique center of the smallest enclosing circle (SEC) of all robots, and then concurrently move there, finishing in one single round).

The Problem (2)

This notion implies that the visibility graph of the robots is a unit disk graph (UDG), i.e. the intersection graph (a graph that represents the pattern of intersections of a family of sets) of a family of unit disks in the Euclidean plane.



The Problem (3)

- The UDG of the robots must be connected initially, meaning that there is a path from any robot to any other robot just following the visibility neighborhoods.
- Robots are anonymous (they do not have unique IDs). If robots have unique IDs, the problem becomes much simpler: they just have to agree on meeting at the location of the robot with the minimum ID.

Question: Are the robots able to meet at a single point and how long does it take to do so?
Answer: Yes, in $\Theta\left(n^2\right)$ rounds (n: number of nodes)
General Idea: in each round, every robot moves to the center of the SEC of the robots in its viewing range, only constrained by the condition that robots must not lose visibility to their neighboring robots.

Model Definition (Problem description and notation (1))

- Set R of n robots $r_1, \ldots, r_n$ in the Euclidead plane
- Goal: gather all robots in one point
- Each robot is represented as a singular point in the plane (this means that robots cannot block each other's views or paths)
- We use a discrete, synchronous time model: in each round t, $t \in \mathbb{N}_0$, all robots act synchronously at the same time
- Positions $p_1(t), \cdots, p_n(t)$ of the robots at the beginning of round t: *configuration* at time t (when the round t under consideration is clear from the context, we will sometimes indentify a robot r_i with its position $p_i(t)$
- Configuration at time 0: *start configuration*
- Time t: the beginning of round t

Model Definition (Problem description and notation (2))

- $d(p_i(t), p_j(t))$ or $d(r_i, r_j)$: the euclidean distance between two robots r_i, r_j
- $d(r_i, r_j) \leq 1$: the robots r_i, r_j can see each other (neighbors). Distance 1 is the *viewing range* of the robots.
- $N_t(r_i)$ or $N(r_i)$ (if time is clear from the context): the set of all neighbors of robot r_i (its neighborhood)
- $UDG_t = (R, E_t)$: a unit disk graph (*visibility graph*) where $(r_i, r_j) \in E_t$ iff r_i and r_j are mutually visible at time t, i.e $d(r_i(t), r_j(t)) \leq 1$

Model Definition (Problem description and notation (3))

We measure the quality of the algorithm by counting the number of synchronous rounds until the robots have gathered in one point. During each round, the robots act according to the Look-Compute-Move (LCM) model:

- First all robots synchronously observe their environment and determine the positions of their neighbors relative to their own position (Look-operation).
- During the Compute-operation, they use the observed positions as input for the algorithm described later.
- The algorithm outputs the point to which the robots move during the following Move-operation.

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

Model Definition (Robot model)

Our robots:

- Have a limited viewing range
- Are oblivious (they don't have memory)
- Don't communicate
- Don't use a common coordinate system
- Are anonymous (they cannot be distinguished from one another)

On the other hand:

- They are able to measure positions of neighbors relative to their own position accurately
- They can compute geometric properties
- They can occupy the same position as other robots

The Algorithm (1)

The algorithm works as follows:

- First, r_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 is used to compute a point x on the line segment between $p_i(t)$ and $c_i(t)$ to which r_i finally moves.
  - For each neighbor r_j, r_i computes **i)** the midpoint m_j between their positions and **ii)** the *limit circle* D_j with center m_j and radius $1/2$.
  - As long as both r_i and r_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 $p_i(t)$ and $c_i(t)$ that maximizes the distance that r_i moves under the constraint that r_i does not leave the circle D_j for any neighbor r_j.

The Algorithm (2)

Algorithm 1 Algorithm of robot r_i in round t

```
{compute target point}
R_i(t) := {all robots visible from r_i including r_i itself}
C_i(t) := smallest enclosing circle of R_i(t)
c_i(t) := center of C_i(t)
{keep connectivity}
∀ r_j ∈ R_i(t) : m_j := midpoint between p_i(t) and p_j(t)
∀ r_j ∈ R_i(t) : D_j := circle with radius 1/2 around m_j
seg := line segment ‾p_i(t), c_i(t)‾
A := ⋂_{r_j ∈ R} D_j ∩ seg
x := point in A that minimizes d(x, c_i(t))
{Note that A ≠ ∅, since p_i(t) ∈ A}
p_i(t+1) := x
```

The Algorithm (3)

Lemma 1

If two robots are neighbors in UDG_t at time t, then they are still neighbors in UDG_{t+1}. In particular, if UDG_0 is connected, then UDG_t is connected for all $t \geq 0$.

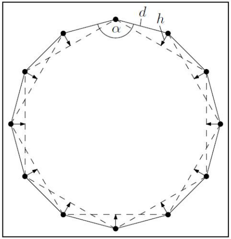
- A robot r_j *hinders* another robot r_i from reaching some point p on the line segment between $p_i(t)$ and $c_i(t)$, if r_i would leave D_j when moving to p.
- If in any round, two robots move to the exact same point, they will stay at a common point for the rest of the execution of the algorithm, because they see the same neighborhood and hence behave exactly the same. We call such robots to have *merged*.

It is proven [2] that this algorithm gathers the robots in one point within finite time.

The Lower Bound (1)

Theorem 1

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



initial distance between two neighboring robots $= 1$
initial circumference of the circle $\approx n$
radius $\approx \frac{n}{2\pi}$

The Lower Bound (2)

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.

The distance that r moves is the height h of the equilateral triangle formed by r and its two neighbors.

- $h = \cos\left(\frac{\alpha}{2}\right) d$
- 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 $\cos\left(\frac{\alpha}{2}\right) \leq -\frac{\alpha}{2} + \frac{\pi}{2}$, resulting in $h \leq \left(-\frac{\alpha}{2} + \frac{\pi}{2}\right) d$
- 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, $h \leq \left(-\frac{\alpha}{2} + \frac{\pi}{2}\right) d \leq \left(-\left(\frac{\pi}{2} - \frac{\pi}{n}\right) + \frac{\pi}{2}\right) d = \frac{\pi}{n} d \leq \frac{\pi}{n}$

The Lower Bound (3)

This means that 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 $\frac{1}{3}n$.

circumference $= 2\pi \times$ *radius*, so 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 $\frac{2}{3}n$. □

The Upper Bound (Geometric Prerequisites (1))

Proposition 1

Let C be the smallest enclosing circle (SEC) of a point set S. Then either

1. there are two points $P, Q \in S$ on the circumference of C such that the line segment $\overline{PQ}$ is a diameter of C, or
2. there are three points $P, Q, R \in S$ on the circumference of C such that the center c of C is inside $\triangle PQR$, which means that $\triangle PQR$ is acute-angled.

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

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

The Upper Bound (Geometric Prerequisites (2))

Now we can define how we measure progress. We will use two progress measures:

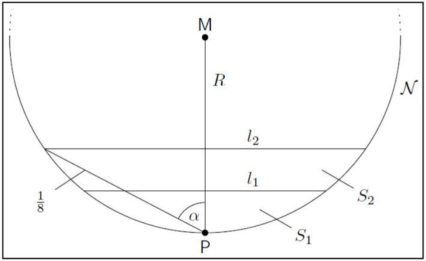
- counting the number of rounds in which robots merge. We have n robots in the beginning, so there can be at most $n-1$ such rounds.
- the algorithm is deterministic and it has been proven that the robots gather in finite time, so we know that for a given start configuration, the point where the robots gather is fixed. We will call this point the *gathering point* M. We define a circle N_t with center M and radius R_t for a round t, such that N_t contains all robots in round t and its radius is minimal. 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 and we use it as a second progress measure.

The Upper Bound (Geometric Prerequisites (3))

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 UDG_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 merge, R_t decreases by at least $\Omega\left(\frac{1}{n}\right)$.
Using these two progress measures, with $R_0 \leq n-1$ and at most $n-1$ rounds in which robots merge, it follows directly that the robots gather in $O\left(n^2\right)$ rounds.

The Upper Bound (Geometric Prerequisites (4))

We consider an arbitrary but fixed round t_0.
Let $N := N_{t0}$ and $R := R_{t0}$.



The intersection points of l_2 and the circle N are in distance $1/8$ from P.
The length of l_2 is bounded by $1/4$.
The main idea of the analysis is to show that in round t_0 and $t_0 + 1$, either two robots merge or all robots leave S_1: this leads to the desired number of rounds.

The Upper Bound (Geometric Prerequisites (5))

Geometrical lemmas for the position of the center of a SEC.
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 2

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

If a robot can only see one single robot in $S_1 \cup S_2$, it cannot compute a target point in S_1.

Lemma 3

The center of the SEC of a non-empty point set $S \subseteq N \backslash (S_1 \cup S_2)$ and a point $A \in S_1 \cup S_2$ cannot lie in the segment S_1.

Lemma 4

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

The Upper Bound (Gathering Algorithm Analysis (1))

Using these geometrical lemmas, we can 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 cannon be hindered from leaving S_1 and S_2.

Lemma 5

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.

Lemma 6

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

The Upper Bound (Gathering Algorithm Analysis (2))

With all these prerequisites, we can now show that if no robots merge, S_1 is empty after two rounds.
First, we analyse the behavior of some robots in round t_0.

Lemma 7

Let 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.

Using this lemma, we prove the following:

Lemma 8

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

The Upper Bound (Gathering Algorithm Analysis (3))

Lemma 9

If $R_t < \frac{1}{2}$, the robots have gathered at one point in round $t+1$.

Theorem 2

The robots gather within $O(n^2)$ *rounds.*

Proof.

Fix an arbitrary round $t_0 \geq 0$. Since lemma 8 holds for any point on the boundary of N_{t_0}, after two rounds either two robots have merged or all robots must be in distance greater than the height of S_1 from the boundary of N_{t_0}. According to lemma 4, the height of S_1 is at least $\frac{1}{128R_t}$ and thus if the robots do not merge, the radius decreases by at least $\frac{1}{128R_t}$, giving that $R_{t+2} \leq R_t - \frac{1}{128R_t} \leq R_t - \frac{1}{128R_0}$. It follows that after $2 \times 128 \times (R_0)^2$ rounds without merging robots, the radius must be less than $\frac{1}{2}$. Now it takes one round to gather the robots (lemma 9). Moreover, since UDG_0 is connected, $R_0 \leq n$. There are at most $n-1$ rounds in which robots merge. The total number of rounds is therefore at most $256n^2 + n$. □

Bibliography

1. Degener, B., Kempkes, B., Langner, T., Meyer auf der Heide, F., Pietrzyk, P., Wattenhofer, R.: A tight runtime bound for synchronous gathering of autonomous robots with limited visibility.
2. Hideki Ando, Yoshinobu Suzuki, and Masafumi Yamashita. Formation and agreement problems for synchronous mobile robots with limited visibility.