The Mutual Visibility Problem for Oblivious Robots

Giuseppe Antonio Di Luna* Paola Flocchini† Federico Poloni‡ Nicola Santoro§ Giovanni Viglietta†

Abstract

Consider a finite set of identical entities, called *robots*, which can move freely in the Euclidean plane. Let $p(t)$ denote the location of robot p at time t; a robot p can see robot q at time t if at that time no other robot lies in the line segment $p(t)q(t)$. We consider the basic problem called Mutual Visibility: starting from arbitrary distinct locations, within finite time the robots must reach, without collisions, a configuration where they all see each other. This problem must be solved by each entity autonomously executing the same algorithm. We study this problem in the standard model of semi-synchronous oblivious robots.

The extensive literature on computability in such a model has never considered this problem because it has always assumed that three collinear robots are mutually visible. In this paper we remove this assumption, and present an algorithm that solves Mutual Visibility. To prove its correctness, we solve a seemingly unrelated problem, Communicating Vessels, which is interesting in its own right. As a byproduct of our solution, we also solve a classical problem for oblivious robots, Near-Gathering, even if one robot is faulty and unable to move.

1 Introduction

Model and previous work. Consider a set of n mobile computational entities, called *robots*, located in the Euclidean plane, each at a distinct point. The robots are anonymous, indistinguishable, without any direct means of communication; each robot is provided with its own local coordinate system (possibly different from the other robots' systems). They operate in rounds, however not all robots are necessarily active at all rounds. In a round, each active robot determines the position (in its own coordinate system) of the other robots, it performs some local computation to determine a destination point, and it moves to this point. The choice of which robots are activated in each round is made by an adversary, but each robot is activated infinitely often. The robots are oblivious: at the end of a round, they forget all their observations and computations. This model of semi-synchronous oblivious mobile robots has been extensively studied, for instance, in [2, 5, 7, 8, 10, 11]. All these investigations share the assumption that visibility is unobstructed; that is, three collinear robots are mutually visible. Not much is known on computing when the visibility of these robots is *obstructed* by the presence of other robots; that is, if two robots p and q are located at $p(t)$ and $q(t)$ at round t, they can see each other if and only if no other robot lies in the segment $p(t)q(t)$ at that time. In fact, the few studies on obstructed visibility have been done in different models: the model of robots in the one-dimensional space $\mathbb{R}^1$ (see [3]); the model of *robots with lights*, where robots have bounded direct communication capabilities and persistent memory (see [6]); and the so-called *fat robots* model, where robots are not points but unit discs, collisions are allowed and can be used as an explicit computational tool (e.g., [1, 4, 8]). Note that, in our model, because robots are oblivious and anonymous and execute the same protocol, if $p(t) = q(t)$ (a collision), then the activation adversary can force $p(t') = q(t')$ for all $t' > t$. Thus, unless this is the intended outcome, collision avoidance is always a requirement for all algorithms in our model.

Our contribution. In this paper we start the investigation of computing with obstructed visibility, and consider the most basic problem, Mutual Visibility, where the robots start from arbitrary distinct positions in the plane and must reach a configuration when they all see each other. This problem is clearly at the basis of any subsequent task requiring complete visibility.

In Section 3.1 we present an algorithm that solves Mutual Visibility without collisions. Each of the n robots moves independently from the others, following its local algorithm, until the robots form a convex n-gon, in spite of the decisions of the scheduling adversary. At this point they all see each other, so the problem is solved and the robots may terminate the execution. Moreover, during this process, they never move outside the convex hull of the robots' original positions (the only exception being when they are initially all collinear). Not exceeding the original convex hull is a desirable feature, e.g., if the robots are initially delimiting an area that is unsafe outside its border. The algorithm works without any agreement among the robots on a common unit of

*Università degli Studi di Roma "La Sapienza", Italy, diluna@dis.uniroma1.it

†University of Ottawa, Canada, flocchin@site.uottawa.ca, viglietta@gmail.com

‡Università di Pisa, Italy, fpoloni@di.unipi.it

§Carleton University, Canada, santoro@scs.carleton.ca

distance, or North direction, or handedness. Actually, these parameters may even change, from activation to activation, for the same robot.

To prove the correctness of our algorithm, in Section 3.3 we solve a seemingly unrelated problem, Communicating Vessels, which is interesting in its own right.

As a byproduct of our solution, we also solve a classical problem for oblivious robots: *collision-less convergence to a point* or Near-Gathering (see [8, 9]). In fact, we show that, if the robots continue to follow our algorithm once they reach full visibility, the convex hull of their positions converges to a point, and the robots approach it without colliding.

Finally, we observe an interesting property of fault-tolerance of our algorithm: if a single robot is faulty and becomes unable to move, the robots will still solve Near-Gathering, converging to the faulty robot's location.

2 Definitions

We use the standard model of semi-synchronous oblivious robots (e.g., see [8]). Let $\mathcal{R} = \{r_1, r_2, \cdots, r_n\}$ be a set of autonomous oblivious mobile robots operating in the Euclidean plane. We denote by $r_i(t) \in \mathbb{R}^2$ the position occupied by robot $r_i \in \mathcal{R}$ at time $t \in \mathbb{N}$; these positions are expressed here in a global coordinate system, which is used for description purposes, but is unknown to the robots.

We say that robot r_i *sees* robot r_j at time t if and only if the line segment $r_i(t)r_j(t)$ does not contain any other robot at that time. Two robots r_i and r_j are said to *collide* at time t if $r_i(t) = r_j(t)$.

Each robot is provided with its own local coordinate system centered in itself, and its own unit of distance. However, there might be no agreement among different robots on the coordinate system, on its handedness, or on the unit of distance; moreover, the coordinate system of a robot might not be preserved over time and might be modified by an adversary.

The robots are anonymous (i.e., without internal identifiers), indistinguishable (i.e., without external markings), without any direct means of communication. Each robot is provided with a private copy of the same algorithm, which it executes locally every time it is activated. At each time instant, a robot is either active or inactive. If active at time $t \in \mathbb{N}$, a robot performs a *Look–Compute–Move* sequence of operations: it determines the positions, in its own coordinate system, of the visible robots (*Look*); using these positions and the value of n as input, the robot executes the algorithm to determine a destination point (*Compute*); finally, the robot moves to the computed destination, if it is different from its current location (*Move*). This sequence of operations is executed “atomically”, and ends by time $t + 1$. The choice of which robots are active at time t is made by an adversary, called *scheduler*, subject only to the fairness restriction that each robot be activated infinitely often. We stress that, at any given round, the scheduler may activate any subset of robots, from the empty set to all of $\mathcal{R}$. A robot can also decide to *terminate* its execution during a Compute phase. When a robot has terminated, it can never move again.

3 Solving the Mutual Visibility problem

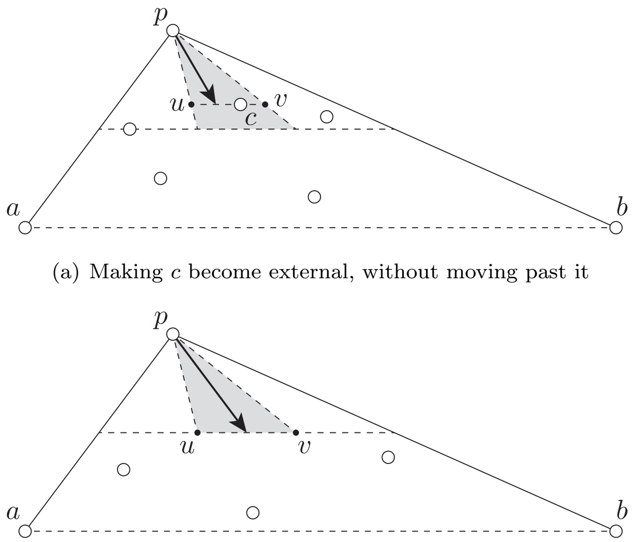
The Mutual Visibility problem is solved if the robots reach a configuration in which they have all terminated their execution, and no three of them are collinear. In other words, all the robots must be mutually visible and in n distinct locations (assuming that $n \geqslant 3$). Such a configuration must be reached from any initial configuration in which the robots' positions are all distinct (this is a necessary condition, as noted in Section 1), and regardless of the activation pattern decided by the scheduler. In the following we present Algorithm 1, which solves Mutual Visibility.

3.1 Algorithm description

Let us consider the convex hull of the robots' locations, at a given time. The robots lying on its boundary are called *external* robots, while the ones lying in its interior are the *internal* robots. The main idea of Algorithm 1 is to make only the external robots move, so to shrink the convex hull. When a former internal robot becomes external, it starts moving as well. Eventually, all the robots become external, and at this point they all see each other. Since the robots know n, they recognize this situation, and they can terminate.

Observe that a robot may not know where the convex hull's vertices are located, because they may be obstructed by other robots. However, it can at least determine whether it is an external or an internal robot. Indeed, being an internal robot is equivalent to being in the interior of the convex hull of the *visible* robots.

If an active robot r_i, located at p, realizes that it is internal, it does not move. Otherwise, it locates its clockwise and counterclockwise neighbors on the convex hull's boundary, say located at a and b, which are necessarily visible. Then, r_i attempts to move somewhere in the triangle pab, in such a way to shrink the convex hull, and possibly make its boundary acquire one or more new robots. To avoid collisions with other robots that may be moving at the same time, r_i's movements are restricted to a smaller triangle, shaded in gray in Figure 1. Moreover, to avoid becoming an internal robot, r_i does not cross any line parallel to ab that passes through another robot, as shown in Figure 1(a). In particular, if no such line intersects the gray area, r_i makes a *default move*, and it moves halfway toward the midpoint of the segment ab, as indicated in Figure 1(b).



(a) Making c become external, without moving past it

(b) Default move

Figure 1: Move of an external robot, in two different cases. Robots' locations are indicated as small circles.

The above rules are sufficient to solve the Mutual Visibility problem in most cases, but there are some exceptions. It is easy to see that there are configurations in which Mutual Visibility is never solved until an internal robot moves, regardless of the algorithm employed. For instance, suppose that the configuration is centrally symmetric, with one robot lying at the center. Let the local coordinate systems of any two symmetric robots be oriented symmetrically and have the same unit distance, and assume that the scheduler chooses to activate all robots at every turn. Then, every two symmetric robots have symmetric views, and therefore they move symmetrically. If the central robot—which is an internal robot—never moves, then the configuration remains centrally symmetric, and the central robot always obstructs all pairs of symmetric robots. Hence Mutual Visibility is never solved, no matter what algorithm is executed.

It turns out that our rules can be fixed in a simple way to resolve also this special case: if there is only one internal robot, it moves to the midpoint of any edge of the convex hull. Note that the only internal robot necessarily sees all the others, and therefore it can identify the configuration, due to its knowledge of n.

Finally, the configurations in which all the robots are collinear need special handling. In this case it is impossible to solve Mutual Visibility unless some robots leave the current convex hull. In Algorithm 1, if a robot sees only one other robot, it realizes that all robots lie on a line, and that it occupies an endpoint. Therefore, it moves orthogonally to that line. When this is done, the previous rules apply.

Algorithm 1: Solving the Mutual Visibility problem

```
Input:
V: set of locations of the robots visible to me
(myself included) expressed in a coordinate system
centered at my location;
n: total number of robots (both visible and
invisible).
p ⟵ (0, 0)
H ⟵ convex hull of V
if H has n non-degenerate vertices then
    Terminate
else if |V| = 2 then
    a ⟵ location of the other visible robot
    Move orthogonally to pa by any amount
else if I lie on the boundary of H then
    a ⟵ my ccw neighbor on the boundary of H
    b ⟵ my cw neighbor on the boundary of H
    γ ⟵ 1/2
    if p ∉ ab then
        foreach c ∈ V \ {p} do
            Let α, β be such that c = α · a + β · b
            if α + β < γ then γ ⟵ α + β
    u ⟵ γ · (2a + b)/3
    v ⟵ γ · (a + 2b)/3
    Move to the midpoint of any connected
    component of uv \ (V \ {p})
else if H has n − 1 non-degenerate vertices then
    Move to the midpoint of any edge of H
```

3.2 Correctness: invariants

In the following we discuss some basic invariants, which will serve to prove the correctess of Algorithm 1. Let $\mathcal{H}(t)$ denote the convex hull of $\{r_1(t), r_2(t), \cdots, r_n(t)\}$.

Suppose that, for some $t \in \mathbb{N}$, $\mathcal{H}(t)$ is not a line segment: the situation is illustrated in Figure 2. Every external robot is bound to remain in the corresponding gray triangle, and by construction all such triangles are disjoint. Moreover, if there is only one internal robot and it is activated, it moves to the midpoint of an edge of $\mathcal{H}(t)$, which does not lie in any gray triangle. It follows that, no matter which robots are activated at time t, they will not collide. Also, $\mathcal{H}(t+1) \subseteq \mathcal{H}(t)$.

Observe that a robot $r \in \mathcal{R}$ is external at time t if and only if there is a half-plane bounded by a straight line through $r(t)$ whose interior contains no robots at time t. Now, referring to Figure 1, it is clear that a robot that is external at time t will also be external at time $t+1$. Indeed, if $r(t) = p$, r will stop at the first horizontal line that contains a robot, or it will make a default move. Therefore, at time $t+1$, r will be found on the line uv, and no robot will lie above this line.

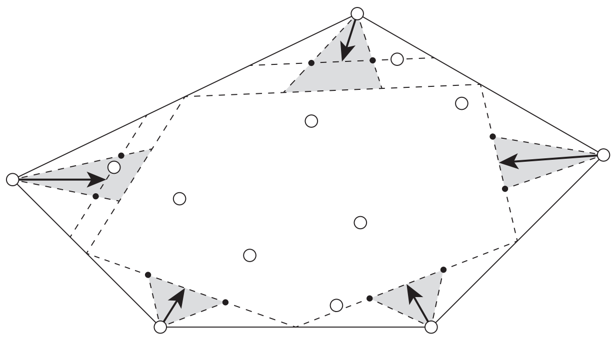


Figure 2: Combined motion of all external robots.

Additionally, if no new robots become external between time t and $t+1$, then the ordering of the external robots around the convex hull is preserved from time t to time $t+1$. This easily follows from the fact that every robot remains in its own gray triangle (cf. Figure 2).

3.3 Correctness: convergence

We seek to prove that Algoritm 1 makes every robot eventually become external. As it will be apparent in the proof of Theorem 6, the crux of the problem is the situation in which only default moves are made (cf. Figure 1(b)). In Lemma 5 we will prove that in this case, if no new robots become external or terminate, all the robots converge to the same limit point. We reduce this sub-problem to the Communicating Vessels problem, as detailed next.

Since we are assuming that only the external robots move, and that their movements depend only on the positions of other external robots, we may as well assume that all robots are external, and that their indices follow their order around the convex hull. Indeed, by the invariants observed in Section 3.2, all robots will remain external throughout the execution, and their ordering around the convex hull will remain the same. So, let r_{i-1}, r_i, r_{i+1} be three external robots, which appear on the boundary of $\mathcal{H}(t)$ consecutively in this order. Let r_i perform a default move at time t. Then, the new position of r_i is a convex combination of the current positions of these three robots, and precisely

$$r_i(t+1) = \frac{r_{i-1}(t)}{4} + \frac{r_i(t)}{2} + \frac{r_{i+1}(t)}{4}. \qquad (1)$$

In general, as different sets of external robots are activated in several rounds, and nothing but default moves are made, the new location of each robot is always a convex combination of the *original* positions of all the robots, obtained by applying (1) to the set of active robots, at every round. In formulas,

$$r_i(t_0+t) = \sum_{j=1}^{n} \alpha_{i,j,t} \cdot r_j(t_0),$$

with $\alpha_{i,j,t} \geqslant 0$ and $\sum_{j=1}^{n} \alpha_{i,j,t} = 1$, assuming that the robots start making only default moves at time t_0. Let $I = \{1, 2, \cdots, n\}$. We fix $j \in I$, and we let $w_{i,t} = \alpha_{i,j,t} - \alpha_{i-1,j,t}$, where indices are taken modulo n. We claim that

$$\lim_{t\to\infty} w_{1,t} = \lim_{t\to\infty} w_{2,t} = \cdots = \lim_{t\to\infty} w_{n,t} = 0. \qquad (2)$$

If such a claim is true (for all $j \in I$), it implies that the robots get arbitrarily close to each other, as t grows. This, paired with the fact that $\mathcal{H}(t_0+t+1) \subseteq \mathcal{H}(t_0+t)$ for every t, as observed in Section 3.2, allows us to conclude that the robots converge to the same limit point.

Our claim can be generalized and reformulated in the following terms. Suppose that n vessels containing water are arranged in a circle, and there is a pipe between each pair of adjacent vessels, regulated by a valve. At every second, some of the valves are opened and others are closed, in such a way that each of the n valves stays open for infinitely many seconds, in total. If a valve between two adjacent vessels stays open between seconds t and $t+1$, then $1/4$ of the surplus of water, measured at second t, flows from the fuller vessel to the emptier one. Our claim is that the amount of water converges to the same limit in all vessels, no matter how the valves are opened and closed. We call this problem Communicating Vessels.

In this formulation, the amount of water in the i-th vessel at time $t \in \mathbb{N}$ would be our previous $w_{i,t}$. However, here we somewhat abstract from the Mutual Visibility problem, and we consider a slightly more general initial configuration, in which the $w_{i,0}$'s are arbitrary real numbers. We set $v_{i,t} = 1$ if the valve between the i-th and the $(i+1)$-th vessel is open between time t and $t+1$ (indices are taken modulo n), and $v_{i,t} = 0$ otherwise. It is easy to verify that activating robot r_i at time t in our previous discussion corresponds to setting $v_{i,t} = 1$ in the Communicating Vessels formulation.

Let us denote by w_t the vector whose i-th entry is $w_{i,t}$, and let $q_{i,t} = w_{i+1,t} - w_{i,t}$. We first prove an inequality on the Euclidean norms of the vectors w_t. Note that the inequality holds regardless of what assumptions are made on the opening pattern of the valves.

Lemma 1 *For every $t \in \mathbb{N}$,*

$$\|w_t\|^2 - \|w_{t+1}\|^2 \geqslant \frac{1}{4} \sum_{i=1}^{n} v_{i,t} \cdot q_{i,t}^2. \qquad (3)$$

Proof. For brevity, let $a = w_{i-1,t}$, $b = w_{i,t}$, $c = w_{i+1,t}$; hence, $q_{i-1,t} = b - a$ and $q_{i,t} = c - b$.

Suppose first that $v_{i-1,t} = v_{i,t} = 1$, i.e., both valves connecting the i-th vessel with its neighbors are open. Then, $w_{i,t+1} = (a + 2b + c)/4$. We have

$$\frac{w_{i-1,t}^2}{4} + \frac{w_{i,t}^2}{2} + \frac{w_{i+1,t}^2}{4} - w_{i,t+1}^2 \geqslant \frac{q_{i-1,t}^2}{8} + \frac{q_{i,t}^2}{8}, \quad (4)$$

which can be obtained by dropping the term $(a-c)^2/16$ from the algebraic identity

$$\frac{a^2}{4}+\frac{b^2}{2}+\frac{c^2}{4}-\frac{(a+2b+c)^2}{16}=\frac{(a-b)^2}{8}+\frac{(b-c)^2}{8}+\frac{(a-c)^2}{16}.$$

Now, suppose instead that $v_{i-1,t}=1$ and $v_{i,t}=0$. Then we have $w_{i,t+1}=(a+3b)/4$, and

$$\frac{w_{i-1,t}^2}{4}+\frac{3w_{i,t}^2}{4}-w_{i,t+1}^2=\frac{3q_{i-1,t}^2}{16}\geqslant\frac{q_{i-1,t}^2}{8}, \qquad (5)$$

where the first equality comes from the identity

$$\frac{a^2}{4}+\frac{3b^2}{4}-\frac{(a+3b)^2}{16}=\frac{3(a-b)^2}{16}.$$

If $v_{i-1,t}=0$ and $v_{i,t}=1$, an analogous argument gives

$$\frac{3w_{i,t}^2}{4}+\frac{w_{i+1,t}^2}{4}-w_{i,t+1}^2\geqslant\frac{q_{i,t}^2}{8}. \qquad (6)$$

Finally, if $v_{i-1,t}=v_{i,t}=0$, $w_{i,t+1}=w_{i,t}$, and trivially

$$w_{i,t}^2-w_{i,t+1}^2=0. \qquad (7)$$

We sum for each $i\in I$ the relevant inequality among (4), (5), (6), (7), depending on the value of $v_{i-1,t}$ and $v_{i,t}$. Each of the terms $q_{i,t}^2/8$ appears twice if and only if $v_{i,t}=1$, and the coefficients of the terms in $w_{i,t}^2$ sum to 1 for every i, hence we get (3). □

From the previous lemma, it immediately follows that the sequence $(\|w_t\|)_{t\geqslant 0}$ is non-increasing. Since it is also bounded below by 0, it converges to a limit, which we call ℓ. Let $M_t=\max_{i\in I}\{w_{i,t}\}$ and $m_t=\min_{i\in I}\{w_{i,t}\}$. Observe that each entry of w_{t+1} is a convex combination of entries of w_t, hence $(M_t)_{t\geqslant 0}$ is non-increasing and $(m_t)_{t\geqslant 0}$ is non-decreasing. Therefore they both converge, and we let $M=\lim_{t\to\infty}M_t$ and $m=\lim_{t\to\infty}m_t$.

Corollary 2

$$m\leqslant\frac{\ell}{\sqrt{n}}\leqslant M.$$

Proof. For every $t\in\mathbb{N}$, we have

$$nM_t^2\geqslant\sum_{i=1}^{n}w_{i,t}^2=\|w_t\|^2\geqslant\ell^2,$$

which proves the second inequality. As for the first inequality, for every $\varepsilon>0$ and large-enough t, we have $nm_t^2\leqslant\|w_t\|^2\leqslant\ell^2+\varepsilon$. □

For the next lemma, we let $V_i=\{t\in\mathbb{N}\mid v_{i,t}=1\}$.

Lemma 3 *Suppose that $|V_i|=\infty$ for at least $n-1$ distinct values of $i\in I$. Then,*

$$M=m=\frac{\ell}{\sqrt{n}}.$$

Proof. Due to Corollary 2, it is enough to prove that $M-m=0$. By contradiction, assume $M-m>0$, and let $\delta=(M-m)/(n+1)>0$. We have

$$\lim_{t\to\infty}\left(\|w_t\|^2-\|w_{t+1}\|^2\right)=\ell^2-\ell^2=0,$$

hence there exists $T\in\mathbb{N}$ such that $\|w_t\|^2-\|w_{t+1}\|^2<\delta^2/4$ for every $t\geqslant T$. By Lemma 1,

$$\frac{q_{i,t}^2}{4}\leqslant\|w_t\|^2-\|w_{t+1}\|^2<\frac{\delta^2}{4}$$

for every $t\geqslant T$ and every i such that $v_{i,t}=1$. This implies $|q_{i,t}|<\delta$, that is, a necessary condition for the valve between the i-th and the $(i+1)$-th vessel to be open at time $t\geqslant T$ is that $|w_{i+1,t}-w_{i,t}|<\delta$. Consider now the $n+1$ open intervals

$$(m,m+\delta),(m+\delta,m+2\delta),\cdots,(m+n\delta,M),$$

each of width δ. Since $M_T\geqslant M$ and $m_T\leqslant m$, there are $w_{i,T}$'s above and below all these intervals. Moreover, by the pigeonhole principle, at least one of the intervals contains no $w_{i,T}$'s, for any $i\in I$. In other words, we can find a partition $I_1\cup I_2=I$, with I_1 and I_2 both non-empty, and a threshold value λ such that $w_{i,T}\leqslant\lambda$ for every $i\in I_1$, and $w_{i,T}\geqslant\lambda+\delta$ for every $i\in I_2$. Hence, at time T, only valves between entries of w_t whose indices belong to the same I_k can be open. It is now easy to prove by induction on $t\geqslant T$ the following facts:

- $\max_{i\in I_1}\{w_{i,t}\}\leqslant\lambda$,
- $\min_{i\in I_2}\{w_{i,t}\}\geqslant\lambda+\delta$,
- $v_{i,t}=0$ whenever i and $i+1$ belong to two different classes of the partition.

Since I_1 and I_2 are non-empty, there must be at least two distinct indices $i'\in I_1$ and $i''\in I_2$ such that $i'+1\in I_2$ and $i''+1\in I_1$ (where indices are taken modulo n). It follows that the i'-th and i''-th valve are never open for $t\geqslant T$, and this contradicts the hypothesis that $|V_i|<\infty$ for at most one choice of $i\in I$. □

This solves the Communicating Vessels problem.

Corollary 4 *Under the hypotheses of Lemma 3, for every $i\in I$,*

$$\lim_{t\to\infty}w_{i,t}=\frac{\ell}{\sqrt{n}}=\frac{\sum_{j=1}^{n}w_{j,0}}{n}.$$

Proof. By Lemma 3, since $m_t\leqslant w_{i,t}\leqslant M_t$, all the limits coincide. Moreover, the sum of the $w_{i,t}$'s does not depend on t; hence their average, taken at any time, must be equal to the joint limit. □

Let us return to the Mutual Visibility problem, to prove our final lemma.

Lemma 5 *If, at every round, each robot makes a default move (cf. Figure 1(b)) or stays still, and no new robots become external or terminate, then all robots' locations converge to the same limit point.*

Proof. As discussed at the beginning of Section 3.3, this is implied by (2). Recall that $w_{i,0} = \alpha_{i,j,0} - \alpha_{i-1,j,0}$, and hence $\sum_{i=1}^{n} w_{i,0} = 0$. Then, (2) follows immediately from Corollary 4. □

We are now ready to prove our main theorem.

Theorem 6 *Algorithm 1 solves* Mutual Visibility.

Proof. If the initial convex hull is a line segment, it becomes a non-degenerate polygon as soon as one of the endpoints is activated. It is also easy to observe (cf. Figure 2) that, from this configuration, the convex hull may never become a line segment. So the invariants discussed in Section 3.2 apply, possibly after a few initial rounds: no two robots will ever collide, and an external robot will never become internal.

Assume by contradiction that Mutual Visibility is not solved, hence the execution never terminates, and therefore the set of external robots reaches a maximum $\mathcal{E} \subsetneq \mathcal{R}$, say, at time $T \in \mathbb{N}$. If there is only one internal robot, it becomes external as soon as it is activated, due to line 20 of the algorithm, contradicting the maximality of $\mathcal{E}$. Therefore there are at least two internal robots at every time $t \geqslant T$. On the other hand, if an external robot makes a non-default move at any time $t \geqslant T$, a new robot becomes external at time $t+1$. Indeed, referring to Figure 1(a), the line uv passes through $p(t+1)$ and $c(t+1)$, and no robot lies above this line at time $t+1$. Hence c has become a new external robot, which again contradicts the maximality of $\mathcal{E}$.

As a consequence, only default moves are made after time T, and by Lemma 5 the robots converge to the same limit point. But since there are at least two internal robots, this means that at least one of them has to move, implying that it becomes external at some point (by the above assumption, only external robots can move), a contradiction. □

4 Related problems and alternative models

We briefly discuss the Mutual Visibility problem in different robot models, and some applications of Algorithm 1 to related problems.

Forming a convex configuration. As already observed, Algorithm 1 also solves the Convex Formation problem, where the robots have to terminate in a strictly convex position. Moreover, no robot ever crosses the perimeter of the initial convex hull unless, of course, all the robots are initially collinear.

Near gathering. If lines 3, 4, 19, 20 are removed from Algorithm 1, it solves the Near-Gathering problem, even with no knowledge of n. Indeed, if there is only one internal robot, either it will become external, or the other robots will converge to its location. If n is known, the robots can also terminate when they get close enough.

Unreliable moves. Suppose that an adversary has the ability to stop any robot during its Move phase, before it reaches the computed destination, but after it has moved by at least δ towards it. By including δ in our computations in Section 3.3, we can prove that Algorithm 1 solves Mutual Visibility, Convex Formation and Near-Gathering in this model, too.

Fault tolerance. Observe that Lemma 3 requires only $n-1$ valves to be opened infinitely often, as opposed to n. This implies that, if the robots do not terminate, they converge to a point even if one robot is unable to move. Therefore, in the presence of one faulty robot, Algorithm 1 still solves Near-Gathering. (If two robots are faulty, Near-Gathering is unsolvable.) Similarly, Mutual Visibility and Convex Formation are solved in the presence of a faulty robot, provided that it is located on the boundary of the convex hull.

Sequential scheduler. If the scheduler is *sequential*, i.e., it activates only one robot at a time, there is a simple algorithm to solve Mutual Visibility, even with no knowledge of n, and even if the moves are unreliable and two robots are faulty. (If three robots are faulty, Mutual Visibility is unsolvable.) When a robot is activated, it just moves by a small amount, without crossing or landing on any line that passes through two robots that it can currently see (including itself), and then it immediately terminates. Clearly, when a robot moves as described, it becomes visible to all other robots. Hence, when all robots have moved at least once, Mutual Visibility is solved.

References

[1] C. Agathangelou, C. Georgiou, and M. Mavronicolas. A distributed algorithm for gathering many fat mobile robots in the plane. In *Proc. of PODC* 2013, 250–259.

[2] N. Agmon and D. Peleg. Fault-tolerant gathering algorithms for autonomous mobile robots. *SIAM J. Comput.*, 36(1):56–82, 2006.

[3] R. Cohen and D. Peleg. Local spreading algorithms for autonomous robot systems. *Theor. Comput. Sci.*, 399(1,2):71–82, 2008.

[4] J. Czyzowicz, L. Gasieniec, and A. Pelc. Gathering few fat mobile robots in the plane. *Theor. Comput. Sci.*, 410(6,7):481–499, 2009.

[5] X. Défago and S. Souissi. Non-uniform circle formation algorithm for oblivious mobile robots with convergence toward uniformity. *Theor. Comput. Sci.*, 396(1,3):97–112, 2008.

[6] G. Di Luna, P. Flocchini, S. Gan Chaudhuri, N. Santoro, and G. Viglietta. Robots with lights: overcoming obstructed visibility without colliding. In *Proc. of SSS* 2014, to appear.

[7] F. Dieudonné, O. Labbani-Igbida, and F. Petit. Circle formation of weak mobile robots. *ACM Trans. Auton. Adapt. Syst.*, 3(4):16:1–16:20, 2008.

[8] P. Flocchini, G. Prencipe, and N. Santoro. *Distributed Computing by Oblivious Mobile Robots*. Morgan & Claypool, 2012.

[9] L. Pagli, G. Prencipe, and G. Viglietta. Getting close without touching. In *Proc. of SIROCCO* 2012, 315–326.

[10] I. Suzuki and M. Yamashita. Distributed anonymous mobile robots: Formation of geometric patterns. *SIAM J. Comput.*, 28(4):1347–1363, 1999.

[11] M. Yamashita and I. Suzuki. Characterizing geometric patterns formable by oblivious anonymous mobile robots. *Theor. Comput. Sci.*, 411(26,28):2433–2453, 2010.