On the Worst-Case Initial Configuration for Conservative Connectivity Preservation

Daichi Kaino

Department of Computer Science and Engineering
Nagoya Institute of Technology
Aichi, Japan
Email:cke17539@stn.nitech.ac.jp

Taisuke Izumi

Department of Computer Science and Engineering
Nagoya Institute of Technology
Aichi, Japan
Email: t-izumi@nitech.ac.jp

***Abstract*—We consider the system of robots with *limited-visibility*, where each robot can see only the robots within the unit visibility range (*a.k.a.* the unit distance range). In this model, we focus on the inherent cost we have to pay for connectivity preservation in the conservative way (i.e., in any execution, no edge of the visibility graph is deleted). We present a bad configuration with the visibility graph of diameter D for which any conservative algorithm requires $\Omega(D^2)$ rounds to make all robots movable, where D is the diameter of the initial visibility graph. This result implies that we inherently need edge-deletion mechanisms to solve many connectivity-preserving problems (as considered in [1], [2], [5]) within $o(D^2)$ rounds.**

***Keywords*-distributed algorithm; mobile robot; limited visibility; lower bound;**

I. Introduction

Algorithmic studies about autonomous mobile robots is recently emerging in the distributed computing community. In most of those studies, a robot is modeled as a point in a Euclidean plane, and its abilities are quite limited: It is usually assumed that robots are *oblivious* (*i.e.* no memory is used to record past situations), *anonymous* (*i.e.* no ID is available to distinguish two robots), and *uniform* (*i.e.* all robots run the same identical algorithm). In addition, it is also assumed that each robot has no direct means of communication. The communication between two robots is done in an implicit way by having each robot observe its environment, which includes the positions of the other robots.

More challenging settings of algorithmic robotics is the *limited visibility* model [1], [2], [5], where each robot can see only the robots within the unit visibility range (*a.k.a.* the unit distance range). The limited visibility is a practical assumption but makes the design of algorithms quite difficult because it prevents each robot from obtaining the global information about all other robots. Furthermore, it also brings another design issue, called *connectivity preservation* [4]: Oblivious robots cannot use the previous history of their execution. Hence, once some robot r_1 disappears from the visibility range of another robot r_2, r_2 can behave as if r_1 does not exist in the system and vice versa. Since the co-operation between r_1 and r_2 becomes impossible, it follows that completing any task starting from those situations is also impossible. This phenomenon can be formally described by using a *visibility graph*, which is the graph induced by the robots (as nodes) and their visibility relationship (as edges). The requirement we have to guarantee in the limited visibility model is that any task or sub-task in an algorithm must be achieved in the manner that preserves the connectivity of the visibility graph.

A standard approach to achieve the connectivity preservation is that we always disallow the movement of the robots which can bring the deletion of edges in the visibility graph. In this paper we call algorithms adopting this approach *conservative*. Since the deletion of an edge does not necessarily bring the disconnection of the visibility graph, conservative algorithms are overly safe in some sense. On the other hand, surprisingly, every known algorithm for the limited visibility model belongs to the class of conservative algorithms.

The main focus of this paper is to reveal the inherent cost we have to pay for the conservative connectivity preservation. As we stated, the conservativeness property restricts the movement of robots causing the edge deletion of visibility graphs. That kind of movement is characterized by the notion of *blocked locations* [3]. In any conservative algorithm, the robot on a blocked location cannot change its position. A simple example of blocked locations is as follows: A robot r_0 is placed at the origin of the global coordinate system, and r_1, r_2, r_3 are placed on $(-1, 0)$, $(1, 0)$, $(0, 0.1)$ respectively. In this case, a small movement by r_0 (e.g., the movement to $(0.1, 0)$) causes no disconnection of the visibility graph. However, that movement of r_0 causes the deletion of edges (r_0, r_1) and (r_0, r_2), and thus generally r_0 is possible to move but it is not possible in conservative algorithms. Any conservative algorithm must stop the movement of r_0 until r_1 or r_2 gets close to r_0. Thus, at least one extra round is incurred to resolve blocked locations of r_0. This can be seen as an extra cost of the conservative approach. From this observation, a natural question raises up: How much time is necessary to make all robots non-blocked in conservative algorithms? A trivial lower bound for this question is to place n robots at the coordinates $(0, 0), (1, 0), \cdots, (n - 1, 0)$. This configuration obviously requires $\Omega(n)$ rounds for making all robots non-blocked. However, the visibility graph of this configuration has diameter $n-1$. It is not so surprising because we can embed a "long chain" of blocked locations if the visibility graph has a large diameter. More precisely, we can trivially have the configuration satisfying that (1) its visibility graph has diameter D and (2) $\Omega(D)$ rounds are required to make all robots non-blocked. On the other hand, the best known upper bound is $O(D^2)$ rounds for configurations with the visibility graphs of diameter D, which is shown in our prior work[3]. The problem of filling the complexity gap between $\Omega(D)$ and $O(D^2)$ has remained open. The contribution of this paper is to close this gap: We show a bad configuration with the visibility graph of diameter D for which any conservative algorithm requires $\Omega(D^2)$ rounds to make all robots non-blocked. This result implies that we inherently need edge-deletion mechanisms to solve many connectivity-preserving problems (as considered in [1], [2], [5]) within $o(D^2)$ rounds.

II. MODEL

The system consists of n robots, denoted by $r_0, r_1, r_2, \cdots$, r_{n-1}. Robots are *anonymous*, *oblivious* and *uniform*. That is, each robot has no identifier distinguishing itself and others, cannot explicitly remember the history of its execution, and works following a common algorithm independent of the value of n. In addition, no device for direct communication is equipped. The cooperation of robots is done in an implicit manner: Each robot has a sensor device to observe the environment (i.e., the positions of other robots). One robot is modeled as a point located on a two-dimensional space. Observing environment, each robot can see the positions of other robots transcripted in its *local coordinate system*. We assume *limited visibility*: Each robot can see only the robots located within unit distance. Each robot executes the deployed algorithm in *computational cycles* (or briefly *cycles*). At the beginning of a cycle, a robot observes the current environment (i.e., the positions of other robots) and determines the destination point based on the deployed algorithm. Then, the robot moves toward the computed destination. It is guaranteed that each robot necessarily reaches the computed destination at the end of the cycle. As the timing model, we assume fully-synchronous model. In fully synchronous worlds,any execution follows a discrete time $1, 2, 3 \cdots$. At the beginning of each time unit, every robot is activated and performs one cycle. Note that this assumption is stronger than the standard ones such as ATOM[6], but it leads more general results because we consider lower bounds. That is, our argument for the worst cases holds even for full-synchronous systems, and thus it clearly holds for other weaker models.

Throughout this paper, we use the following notations and terminology: To specify the location of each robot consistently, we use the global coordinate system. Notice that the global coordinate system is introduced only for ease the explanation, and thus robots are not aware of it. The origin of the global coordinate system is denotes by $\mathbf{o}$.For any two coordinates $\mathbf{a}$ and $\mathbf{b}$, $\overline{\mathbf{ab}}$ denotes the segment whose endpoints are $\mathbf{a}$ and $\mathbf{b}$, and $|\mathbf{ab}|$ denotes its length. A *configuration* is the multiset consisting of all robot locations. We define $C(t)$ as the configuration at t.

A. Visibility Graph

A visibility graph $G(t)$ is the graph where nodes represent robots and an edge between two robots implies the visibility between two robots(See fig1). More formally, the visibility graph at t consists of n nodes $\{v_0, v_1, v_2, \cdots v_{n-1}\}$. Nodes v_i and v_j are connected if and only if r_i and r_j are visible to each other.

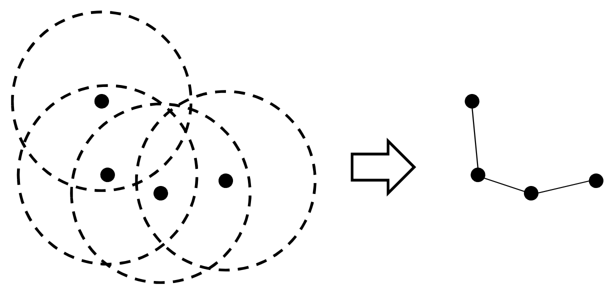

Figure 1. Visibility graph

B. Conservative Connectivity Preservation

The algorithms we consider in this paper belongs to a class called *conservative connectivity preservation*. Formally, an algorithm $\mathcal{A}$ is *conservatively connectivity-preserving* if in any execution of $\mathcal{A}$ edges of the visibility graph are never deleted, That is, let $G(t) = (V, E(t))$ be the visibility graph at t, any execution of $\mathcal{A}$ satisfies $E(t) \subseteq E(t+1)$.

C. Blocked location

We explain the notion of *blocked* locations. Intuitively, a blocked location is the place such that the robot on that location cannot move without deletion of edges in the visibility graph.

Definition 1: Let $\mathbf{p_c}$ be a location, C be the circle centered at $\mathbf{p_c}$ with diameter one, and $B = \{\mathbf{p_0}, \mathbf{p_1}, \mathbf{p_2}, \cdots \mathbf{p_j}\}$ be the set of all locations on the boundary of C. The location $\mathbf{p_c}$ is *blocked* if no arc of C with a center angle less than π can contain all locations in B.
Examples illustrating the notion of blocked locations are shown in Fig. 2, Fig.3.

Intuitively, for a robot r_i to be movable while preserving edges of the visibility graph, its destination must be within distance one from the robots that r_i sees before the

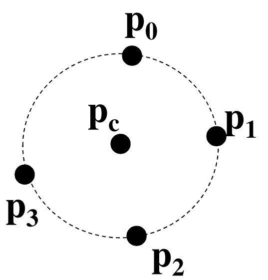

Figure 2. Blocked location

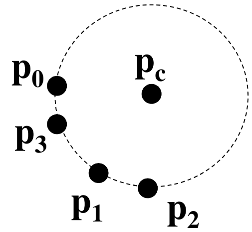

Figure 3. Non-blocked location

movement. For a robot at a blocked location there is no such destination. Assume the contrary, let r_i be blocked and move to some other point $\mathbf{p}(\neq \mathbf{r_i})$. Then, we take the line l which is orthogonal to the vector $\mathbf{p} - \mathbf{r_i}$ and passes through $\mathbf{r_i}$. This line cuts the circle C into two arcs with center angle π. From the definition of blocked points, both arcs have at least one robot. However, the arc in the opposite side of $\mathbf{p}$ (about l) is out of r_i's visibility after the movement to $\mathbf{p}$ (see Fig. 4. Thus, if a robot r_j is on a blocked location, it cannot move anywhere without deletion of edges.

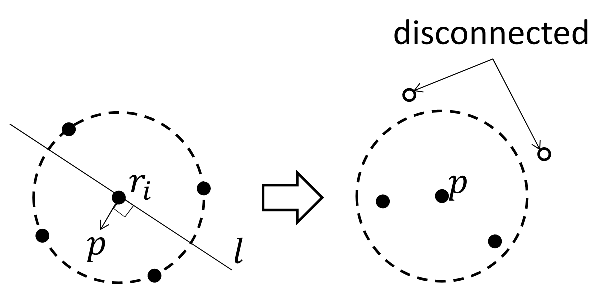

Figure 4. The deletion of edges in visibility graphs

The observation above implies the following lemma:

Lemma 1: In any execution of conservative connectivity-preserving algorithms, no robot on a blocked configuration can not move.

III. Lower Bound

In this section, we show that $\Omega(D^2)$ rounds are necessary to make all robots non-blocked. More precisely,given an arbitrary value D we construct an initial configuration for which any conservative connectivity-preservation algorithm takes $\Omega(D^2)$ rounds to make all robots non-blocked.

A. The Worst-Case Construction

We refer the initial configuration constructed in this section as $C_{BAD}(D)$. For any given $D > 0$. the configuration $C_{BAD}(D)$ consists of $(2 + D)2^{D^2}$ robots (and thus $D = O(\sqrt{\log n})$ must be satisfied).

We define π_k to be the circle of radius $\sqrt{k}$ whose center is at the origin $\mathbf{o}$ of the global coordinate system. For $k = 0$ π_k represents the origin $\mathbf{o}$. The construction is done by recursively placing robots on π_k to block all the robots in π_{k-1}. Let R_k be the set of points on π_k where a robot will be placed.

We first show a fundamental lemma for the construction of R_k $(k \geq 0)$.

Lemma 2: We define $\mathbf{p}$ to be any points on the π_k,and $l_{\mathbf{p}}$ to be the tangent line of π_k at $\mathbf{p}$.Then letting $l_{\mathbf{p}} \cap \pi_{k+1} = \{\mathbf{q}_1, \mathbf{q}_2\}, |\mathbf{q}_1 - \mathbf{p}| = |\mathbf{q}_2 - \mathbf{p}| = 1$ is satisfied.

Proof: Because point $\mathbf{p}$ is placed on π_k, $|\mathbf{p}| = \sqrt{k}$. Similarly,$|\mathbf{q_1}| = |\mathbf{q_2}| = \sqrt{k+1}$. Also hold points $\mathbf{q_1}$ and $\mathbf{q_2}$ are placed on line $l_{\mathbf{p}}$ and thus $\overline{\mathbf{op}}$ is orthogonal to $l_{\mathbf{p}}$. We can apply the Pythagorean theorem to three points $\mathbf{o}$,$\mathbf{p}$,$\mathbf{q_1}$($or\mathbf{q_2}$)(See Fig5). Thus $|\mathbf{q}_1 - \mathbf{p}| = |\mathbf{q}_2 - \mathbf{p}| = 1$ holds. ■

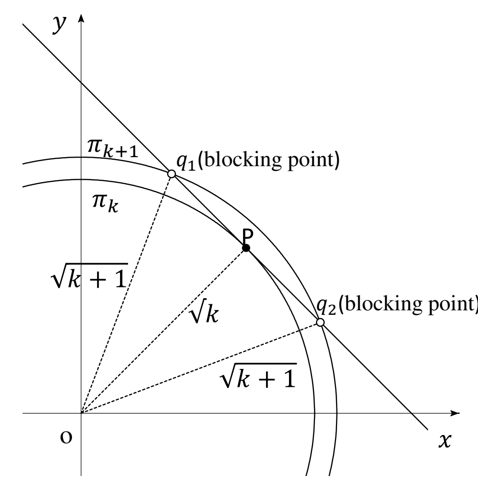

Figure 5. The proof of Lemma 2 and blocking points

The two points $l_{\mathbf{p}} \cap \pi_{k+1}$ defined in this lemma is called the *blocking points* of $\mathbf{p}$. This lemma naturally induces the construction of $C_{BAD}(D)$, which is defined as follows:

- Place one robot at the origin of the global coordinate system (i.e., on π_0). In addition, place two robots at the coordinate $(1, 0)$ and $(0, -1)$. The points in R_1 consists of these two robots.
- For any $\mathbf{p} \in \pi_k$ $(2 \leq k \leq D^2)$ where a robot is located, place two robots on its two blocking points.
- To bound the diameter within D, for each segment between $\mathbf{p} \in R_k$ $(2 \leq k \leq D^2)$ and the origin, place $\lfloor \sqrt{k} \rfloor$ robots with unit interval(See Fig:6).

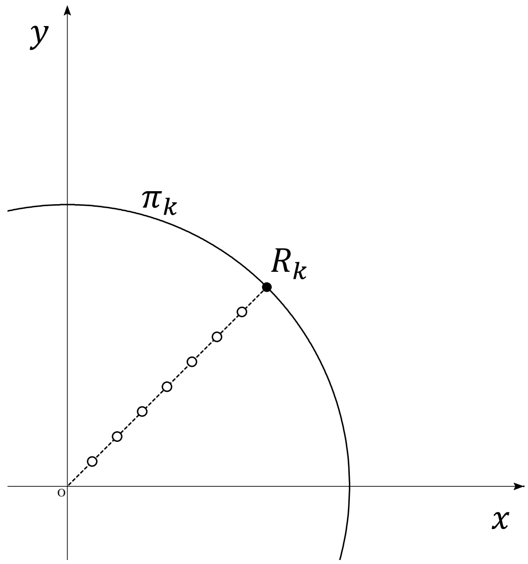

Figure 6. The third step of constructing $C_{BAD}(D)$

Figure 7 shows the construction for $D^2 = 3$. By the third step of the construction above, it is obvious that the visibility graph of $C_{BAD}(D)$ has diameter at most D. Thus remaining issue is to show that this configuration requires D^2 rounds to make all robots non-blocked. We show the example of C_{BAD}.

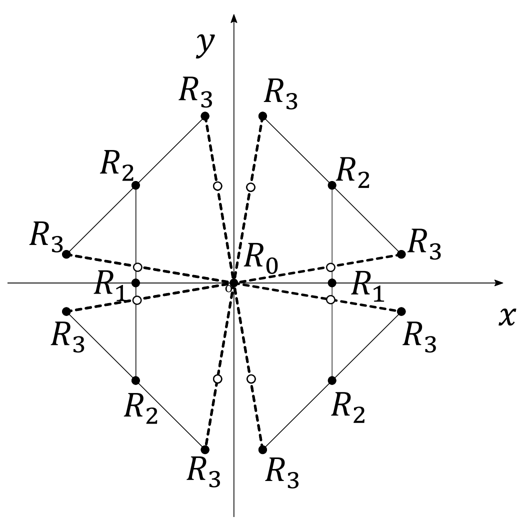

Figure 7. The illustration of C_{BAD} when k=3

Lemma 3: Any robots on R_{D^2-k} are blocked at round k.

Proof: We prove the lemma by induction on k. (**Basis**): If $k = 0$ and 1, the lemma trivially holds. (**Inductive step**): Suppose as the induction hypothesis that all robots in R_{D^2-k} are blocked at round k. In initial placement, any points $\mathbf{p} \in R_{D^2-(k+1)}$ are placed at each blocking points. From including step,robots on the these points blocked until round k, so they don't yet move at beginning time of round $k+1$. Therefor, robots on points $\mathbf{p} \in R_{D^2-(k+1)}$ are blocked at round $k+1$. ■

Consequently, we have the main lemma below:

Theorem 1: For any D, the exists a configuration $C_{BAD}(D)$ with the visibility graph of diameter D such that in any execution of conservative algorithms starting from $C_{BAD}(D)$ requires $\Omega(D^2)$ rounds to make all the robots non-blocked.

This result implies that for many connectivity-preserving problems, no algorithm can achieve the running time sub-quadratic of D at the worst case unless it incurs edge deletions of visibility graphs, which can be seen as an inherent cost by conservative algorithms.

IV. Conclusion

In this paper, we presented a bad configuration with the visibility graph of diameter D for which any conservative algorithm requires $\Omega(D^2)$ rounds to make all robots movable, where D is the diameter of the initial visibility graph. Since we need $(2+D)2^{D^2}$ robots, to construct the bad configuration, our result holds only for the case of $D = O(\sqrt{\log n})$. An open problem is to present the similar bad configuration for larger D (i.e. $D = \omega(\sqrt{\log n})$).

References

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

[2] Paola Flocchini and Giuseppe Prencipe and Nicola Santoro and Peter Widmayer, Gathering of asynchronous robots with limited visibility, Theoretical Computer Science, 337(1-3), pages 147-168, 2005.

[3] Taisuke Izumi and Maria GradinariuPoptop-Butucaru and Sebastien Tixeuil, Connectivity Preserving Scattering of Mobile Robots with Limited Visibility, Proc. of Intl. Symposium on Stabilization, In Safety and Security of Distributed Systems,volume 6336, pages 319-331, 2010.

[4] M. Ji and M. Egerstedt, Distributed Coordination Control of Multi-Agent Systems while Preserving Connectedness, IEEE Transactions on Robotics, volume 23(4), 2007.

[5] Souissi, Samia and Défago, Xavier and Yamashita, Masafumi, Using eventually consistent compasses to gather memory-less mobile robots with limited visibility, ACM Transactions on Autonomous and Adaptive Systems, volume 4(1), pages 1-27, 2009.

[6] Ichiro Suzuki and Masafumi Yamashita,Distributed Anonymous Mobile Robots: Formation of Geometric Patterns,SIAM Journal of Computing,volume 28(4),pages 1347-1363,1999.