International Journal of Networking and Computing – www.ijnc.org, ISSN 2185-2847
Volume 12, Number 1, pages 2–25, January 2022

Visibility-optimal gathering of seven autonomous mobile robots on triangular grids

Masahiro Shibata
Kyushu Institute of Technology
680-4 Kawazu, Iizuka, Fukuoka 820-8502, Japan

Masaki Ohyabu
Nagoya Institute of Technology
Gokiso-cho, Showa-ku, Nagoya, Aichi, 466-8555, Japan

Yuichi Sudo
Hosei University
3-7-2 Kajino-cho, Koganei-shi, Tokyo, 184-8584, Japan

Junya Nakamura
Toyohashi University of Technology
1-1 Hibarigaoka, Tempaku-cho, Toyohashi, Aichi, 441-8580, Japan

Yonghwan Kim
Nagoya Institute of Technology
Gokiso-cho, Showa-ku, Nagoya, Aichi, 466-8555, Japan

Yoshiaki Katayama
Nagoya Institute of Technology
Gokiso-cho, Showa-ku, Nagoya, Aichi, 466-8555, Japan

Received: July 24, 2021
Accepted: September 13, 2021
Communicated by Susumu Matsumae

Abstract

In this paper, we consider the gathering problem of seven autonomous mobile robots on triangular grids. The gathering problem requires that, starting from any connected initial configuration where a subgraph induced by all robot nodes (nodes where a robot exists) constitutes one connected graph, robots reach a configuration such that the maximum distance between two robots is minimized. For the case of seven robots, gathering is achieved when one robot has six adjacent robot nodes (they form a shape like a hexagon). In this paper, we aim to clarify the relationship between the capability of robots and the solvability of the gathering

[0]The conference version of this paper is published in the proceedings of 23rd Workshop on Advances in Parallel and Distributed Computational Models (APDCM 2021).

problem on a triangular grid. In particular, we focus on visibility range of robots. To discuss the solvability of the problem in terms of the visibility range, we consider strong assumptions except for visibility range. Concretely, we assume that robots are fully synchronous and they agree on the direction and orientation of the x-axis, and chirality on the triangular grid. In this setting, we first consider the weakest assumption about visibility range, i.e., robots with visibility range 1. In this case, we show that there exists no collision-free algorithm to solve the gathering problem. Next, we extend the visibility range to 2. In this case, we show that our algorithm can solve the problem from any connected initial configuration. Thus, the proposed algorithm is optimal in terms of visibility range.

Keywords: distributed system, mobile robot, gathering problem, triangular grid

1 Introduction

1.1 Background

Studies for (autonomous) mobile robot systems have emerged recently in the field of Distributed Computing. Robots aim to achieve some tasks with limited capabilities. Most studies assume that robots are uniform (they execute the same algorithm and cannot be distinguished by their appearance) and oblivious (they cannot remember their past actions). In addition, it is assumed that robots cannot communicate with other robots explicitly. Instead, the communication is done implicitly; each robot can observe the positions of the other robots.

1.2 Related work

Since Suzuki and Yamashita presented the pioneering work [1], many problems have been studied in various settings. For example, the gathering problem, which requires all robots to meet at a non-predetermined single point, has been studied in various environments. In the two-dimensional Euclidean space (a.k.a., the continuous model), Suzuki and Yamashita [1] showed that when robots are not fully synchronous, the deterministic gathering of two robots is impossible without additional assumptions. This impossibility result was generalized to an even number of robots initially located evenly at two positions by Courtieu et al. [2] (those configurations are known as *bivalent* configurations). By contrast, Dieudonné and Petit [3] showed that, by adding the assumption that robots can count the exact number of robots at each position (this ability is called the *strong multiplicity detection*), an odd number of robots can gather from any initial position.

The gathering problem in the discrete space (a.k.a., the graph model) has also been studied. In the discrete space, robots stay at fixed positions (the nodes of the graph), and move from one position to the next position through edges of the graph. For (square) grid graphs, D'Angelo et al. [4] considered gathering for robots without multiplicity detection and Castenow et al. [5] considered it for robots without a common compass (a common sense of direction or north, east, south, and west on a grid). For ring graphs, Klasing et al. [6] characterized a set of solvable initial configurations, except for one type of symmetric initial configurations, and in [7] they clarified the feasibility remained open in [6]. D'Angelo et al. [8] proposed an algorithm such that (i) achieves gathering from solvable initial configurations and (ii) detects that gathering is not feasible from unsolvable initial configurations. Stefano and Navarra [9] analyzed the required total number of robot moves to solve the gathering problem in rings. For robots with *local-weak multiplicity detection*, which is a capability for robots to detect whether there exists another one robot or more than one another robot at the current node, Izumi et al. [10] proposed an algorithm to solve the gathering problem from aperiodic and asymmetric initial configurations. Kamei et al. [11] (resp., [12]) proposed an algorithm to solve the problem for an odd (resp., even) number of robots with the local-weak multiplicity detection from symmetric but aperiodic initial configurations.

As a variant of mobile robots, gathering of *fat robots* is considered [13, 14, 15]. Each fat robot dominates a space of a unit disc. There are several definitions of the gathering problem for fat robots, e.g., robots achieve gathering when (i) they form a connected configuration (each robot touches at least one other robot and all robots form one connected formation) or (ii) they reach a configuration

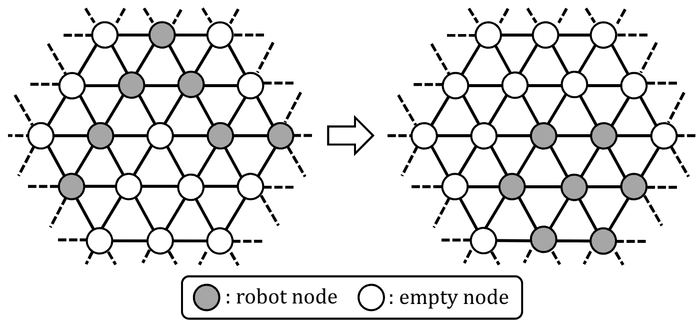

Figure 1: An example of the gathering problem.

such that the maximum distance between two robots is minimized. For both the definitions, a collision is not allowed. Thus, introducing sizes gives several definitions of the gathering problem, which is an interesting point. Czyzowicz et al. [13] considered gathering of (i) for three or four fat robots in the continuous model, and Chrysovalandis et al. [14] studied gathering of (i) for an arbitrary number of fat robots. Ito et al. [15] considered gathering of (ii) on discrete square grids.

Recently, one of computational models for programmable matter, *amoebot* has been introduced [16]. Each amoebot moves on a triangular grid and occupies one or two adjacent nodes. Each amoebot has a finite memory, limited visibility range, and ability to communicate with a robot staying at an adjacent node. Several problems using amoebots have been considered, such as leader election [17], gathering [18], and shape formation (or pattern formation) [19, 20]. Recall that while amoebots have finite memory and communication capability, (standard) autonomous mobile robots have no memory or communication capability. Hence, the mobile robot model is weaker than the amoebot model, and it is interesting to clarify solvability of problems between the mobile robot model and the amoebot model.

Meanwhile, when considering a discrete space, a space filled by regular polygons is sometimes preferable because its simple structure helps to design an algorithm and to discuss the solvability of a problem among various robot models. In addition, (i) only triangular, square, and hexagonal grids are discrete spaces filled by regular polygons, (ii) gathering on a square gird has already been studied [15], and (iii) recently the amoebot model has been extensively studied on a triangular grid. Hence, in this paper we consider gathering of mobile robots on a triangular grid.

1.3 Our contribution

In this paper, we consider the gathering problem of seven mobile robots on triangular grids. We say in this paper that gathering is achieved when robots reach a configuration such that the maximum distance between two robots is minimized. For the case of seven robots, letting a *robot node* be a node where a robot exists, gathering is achieved when one robot has six adjacent robot nodes like Fig. 1. This implies that robots form a (filled) hexagon. In this paper, we aim to clarify the relationship between the capability of robots and the solvability of the gathering problem on a triangular grid. In particular, we focus on visibility range of robots. To discuss the solvability of the problem in terms of the visibility range, we consider strong assumptions except for visibility range. Concretely, we assume that robots are fully synchronous, and they agree on the direction and orientation of the x-axis, and chirality on the triangular grid. In this setting, we first consider the weakest assumption about visibility range, i.e., robots with visibility range 1. In this case, we show that there exists no collision-free algorithm to solve the gathering problem. Next, we extend the visibility range to 2. In this case, we show that our algorithm can solve the problem from any connected initial configuration. Thus, the proposed algorithm is optimal in terms of visibility range.

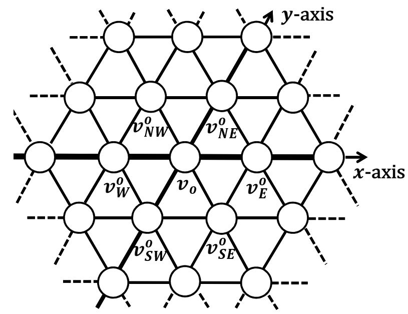

Figure 2: An example of a triangular grid.

2 Preliminaries

2.1 System model

An (infinite) triangular grid is an undirected graph $G = (V, E)$, where V is the set of nodes and E is the set of edges. The grid has one special node called *origin*, and we denote it by v_o. Each node $v_j \in V$ has six *adjacent nodes*: east (v^j_E or E), southeast (v^j_{SE} or SE), southwest (v^j_{SW} or SW), west (v^j_W or W), northwest (v^j_{NW} or NW), and northeast (v^j_{NE} or NE). The axis including v_o and v^o_E (resp., v_o and v^o_{NE}) is called the *x-axis* (resp., *y-axis*)[1]. An example is given in Fig. 2. In addition, a sequence of $k+1$ distinct nodes $(v_0, v_1, \ldots, v_k)$ is called a *path* with length k if $\{v_i, v_{i+1}\} \in E$ for all $i \in [0, k-1]$. The *distance* between two nodes is defined as the length of the shortest path between them.

In this paper, we consider seven mobile robots and denote the robot set by $R = \{r_0, r_1, \ldots, r_6\}$. Robots considered here have the following characteristics. Robots are *uniform*, that is, they execute the same algorithm and cannot be distinguished by their appearance. Robots are *oblivious*, that is, they have no persistent memory and cannot remember their past actions. Robots cannot communicate with other robots directly. However, robots have limited visibility range and they can observe the positions of other robots within the range. This means that robots can communicate implicitly by their positions. We consider two problem settings about robots: *robots with visibility range 1* and *robots with visibility range 2*. Robots with visibility range 1 can observe nodes within distance 1, that is, they can only observe their six adjacent nodes. On the other hand, robots with visibility range 2 can observe nodes within distance 2 (eighteen nodes in total). We assume that they are transparent, that is, even if a robot r_i and several robots exist on the same axis, r_i can observe all the robots on the axis within its visibility range. Robots do not know the position of the origin, but they agree on the direction and orientation of the x-axis, and chirality (the orientation of axes, e.g., clockwise or counter-clockwise) in the triangular gird.

Each robot executes the algorithm by repeating *Look-Compute-Move cycles.* At the beginning of each cycle, the robot observes positions of the other robots within its visibility range (Look phase). According to the observation, the robot computes whether it moves to its adjacent node or stays at the current node (Compute phase). If the robot decides to move, it moves to the node by the end of the cycle (Move phase). Robots are fully synchronous (FSYNC), that is, all robots start every cycle simultaneously and execute each phase synchronously. We assume that a *collision* is not allowed during execution of the algorithm. Here, a collision represents a situation such that two robots traverse the same edge from different directions or several robots exist at the same node. Concretely, the following three behaviors are not allowed: (a) some robot r_i (resp., r_j) staying at node v_p (resp., v_q) moves to v_q (resp., v_p), (b) some robot r_i staying at node v_p remains at v_p and robot r_j staying at node v_q moves to v_p, and (c) several robots move to the same empty node.

A *configuration* of the system is defined as the set of locations of each robot. Here, the location of a robot r_i is defined as the position that (1) r_i is currently staying at and (2) is represented as

[1] Although the origin, the x-axis, and the y-axis are terms of the coordinate system, we use these terms for explanation. In the following, we use several terms of the coordinate system.

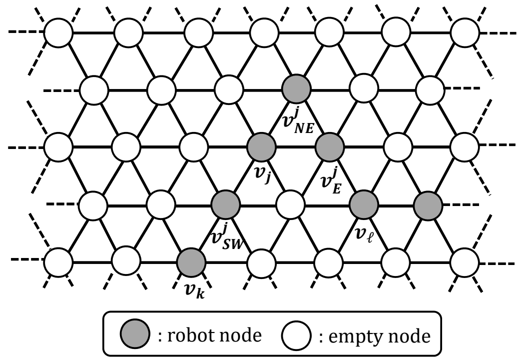

Figure 3: An example of a configuration.

an intersection of an axis parallel to the x-axis and an axis parallel to the y-axis. Each axis ax is represented by (i) whether it is parallel to the x-axis or the y-axis and which direction it is far from the axis, and (ii) the number of axes between the axis including v_o (i.e., the x-axis or the y-axis) and ax. However, robots do not know the position of v_o and they cannot use information of (global) locations. A node is called a *robot node* if the node is occupied by a robot. Otherwise, the node is called an *empty node*. We assume that the initial configuration is connected, that is, the subgraph of G induced by the seven robot nodes is connected. This assumption of connectivity is necessary because, if a configuration becomes unconnected and a robot r has no adjacent robot node, r cannot know the direction to reconstruct a connected configuration due to obliviousness, which implies that robots cannot achieve gathering.

When a robot executes a Look phase, it gets a *view* of the system. A view of a robot is defined as the set of robot nodes within its visibility range. For example, in Fig. 3, a robot at node v_j recognizes that nodes v_E^j, v_{SW}^j, and v_{NE}^j are robot nodes when its visibility range is 1 and recognizes that nodes v_k and v_ℓ are also robot nodes when its visibility range is 2.

2.2 Gathering problem

The gathering problem of mobile robots requires that starting from any connected initial configuration, the robots terminate in a configuration such that the maximum distance between two robot nodes is minimized. In the case of seven robots, gathering is achieved when one robot has six adjacent robot nodes (Fig. 1). Concretely, we define the problem as follows.

Definition 1. *A collision-free algorithm $\mathcal{A}$ solves the gathering problem of seven autonomous mobile robots on a triangular grid if and only if the system reaches a configuration such that one robot has six adjacent robot nodes and no robot moves thereafter, without a collision throughout the execution of $\mathcal{A}$.*

3 Robots with visibility range 1

In this section, for robot with visibility range 1, we show that there exists no collision-free algorithm to solve the problem.

Theorem 1. *For robots with visibility range 1, there exists no collision-free algorithm to solve the gathering problem even in the fully synchronous (FSYNC) model.*

Proof. We show the proof by contradiction, that is, we assume that there exists a collision-free algorithm $\mathcal{A}$ to solve the gathering problem from any connected initial configuration. In the proof, we consider several configurations and robot behaviors, and show that if some robot moves to some direction by Algorithm $\mathcal{A}$, several robots cannot move anywhere (i.e., they have to stay at the current nodes) since a collision occurs or the configuration becomes unconnected. Eventually, we

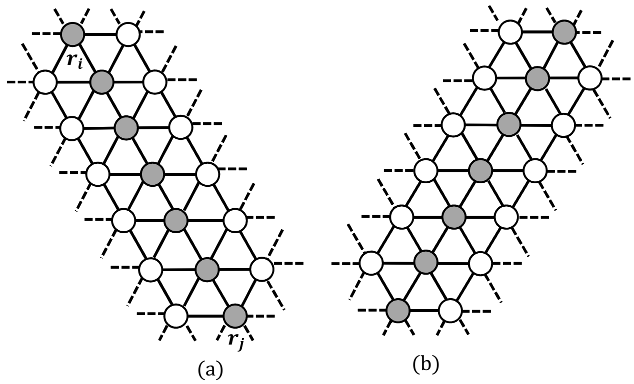

Figure 4: Configurations we consider in the proof.

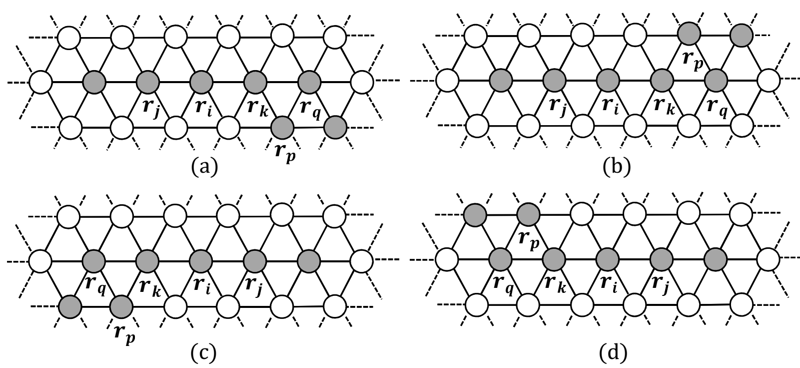

Figure 5: Configurations showing that a robot with two adjacent robot nodes W and E must stay at the current node.

show that there is a configuration such that all robots need to stay at the current nodes and they cannot achieve gathering, which is a contradiction.

First, we consider the configuration of Fig. 4 (a). In the figure, robot r_i (resp., r_j) has one adjacent robot node SE (resp., NW) and the other robots have two adjacent robot nodes SE and NW, respectively. In such a configuration, we first show that intermediate robots cannot leave the current nodes.

Lemma 1. *A robot with two adjacent robot nodes W and E, SW and NE, or NW and SE must stay at the current node.*

Proof. We consider configurations of Fig. 5. In each configuration, robots r_i and r_j have two adjacent robot nodes W and E. On the other hand, robots r_p and r_q have three adjacent robot nodes and they must stay at the current nodes because they cannot detect whether the current configuration is a gathering-achieved configuration or not. In addition, if r_i moves to W, NW, or SW, r_j also moves to the same direction because they have the same view. Then, either in Fig. 5 (a) or (b), wherever r_k moves to, a collision occurs or the configuration becomes unconnected. By a similar discussion, when r_i and r_j move to E, NE, or SE, a collision occurs or the configuration becomes unconnected either in Fig. 5 (c) or (d). Thus, a robot with two adjacent robot nodes E and W cannot leave the current node. By the similar discussion, we can show that a robot with two adjacent robot nodes SW and NE, or NW and SE must stay at the current node. Thus, the lemma follows. □

By this lemma, we can have the following two colloraries.

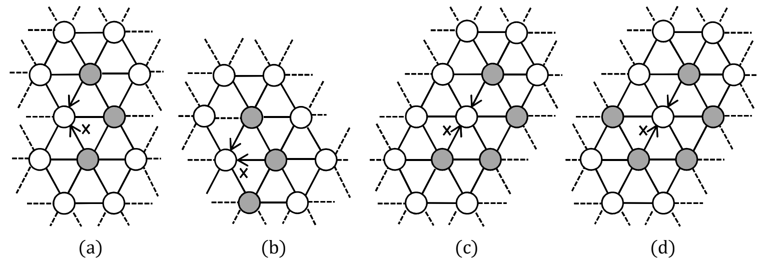

Figure 6: Prohibited behaviors when a robot with one adjacent robot node SE moves to SW.

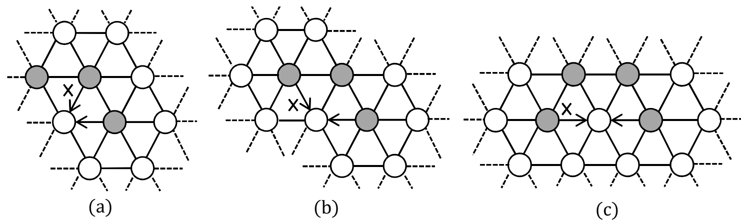

Figure 7: Prohibited behaviors when a robot with one adjacent robot node NW moves to node W.

Collorary 1. *A robot with one adjacent robot node E, SE, SW, W, NW, or NE can move only to NE or SE, E or SW, SE or W, SW or NW, W or NE, or NW or E if it moves, respectively.*

Collorary 2. *A robot with two adjacent robot nodes E and SW, SE and W, SW and NW, W and NE, NW and E, or NE and SE can move only to node SE, SW, W, NW, NE, or E if it moves, respectively.*

By Lemma 1, intermediate robots in Fig. 4 (a) cannot leave the current nodes, and hence r_i or r_j has to leave the current node. Without loss of generality, we assume that in $\mathcal{A}$ robot r_i with one adjacent robot node SE moves to SW. Notice that r_i can move only to SW or E by Collorary 1. In the following, we consider several robot behaviors and eventually show that a robot with one adjacent robot node NE or SW must stay at the current node. Then, in a configuration of Fig. 4 (b), all robots must stay at the current nodes and they cannot solve the gathering problem, which is a contradiction.

When a robot with one adjacent robot node SE moves to SW, several robot behaviors are not allowed since a collision occurs, as shown in Fig. 6 (for simplicity, we omit robot nodes unrelated to prohibited robot behaviors). Concretely, we have the following proposition.

Proposition 1. *When a robot with one adjacent robot node SE moves to SW, the following four robot behaviors are not allowed: (a) a robot with one adjacent robot node NE moves to NW, (b) a robot with two adjacent robot nodes NW and SW moves to W, (c) a robot with one adjacent robot node E moves to NE, and (d) a robot with two adjacent robot nodes NW and E moves to NE.*

In the following, we consider the following five cases: (1) a robot with one adjacent robot node NW moves to W, (2) a robot with one adjacent robot node SW moves to SE, (3) a robot with one adjacent robot node NE moves to E, (4) a robot with one adjacent robot node NW moves to NE, and (5) a robot with one adjacent robot node SW moves to W. In each case, we show that the assumed robot behavior is not allowed. Thus, by Proposition 1-(a) and cases (2), (3), and (5), robots cannot achieve gathering from the configuration of Fig. 4 (b), which is a contradiction (results of cases (1) and (4) are used for cases (2), (3), and (5)).

Case 1: a robot with one adjacent robot node NW moves to W. In this case, as shown in Fig. 7, the following three robot behaviors are not allowed: (a) a robot with two adjacent robot nodes W and SE moves to SW, (b) a robot with one adjacent robot node E moves to SE, and (c) a robot with one adjacent robot node NE moves to E. Then, let us consider the configuration of Fig. 8.

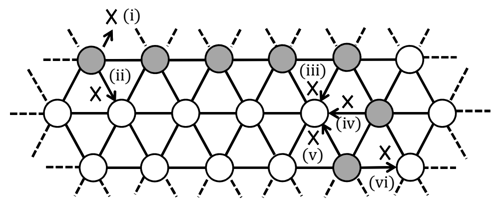

Figure 8: An unsolvable configuration when a robot with one adjacent robot node NW moves to W ((i): by Fig. 6 (c), (ii): by Fig. 7 (b), (iii): by Fig. 7 (a), (iv): by Fig. 6 (b), (v): by Fig. 6 (a), (vi): by Fig. 7 (c)).

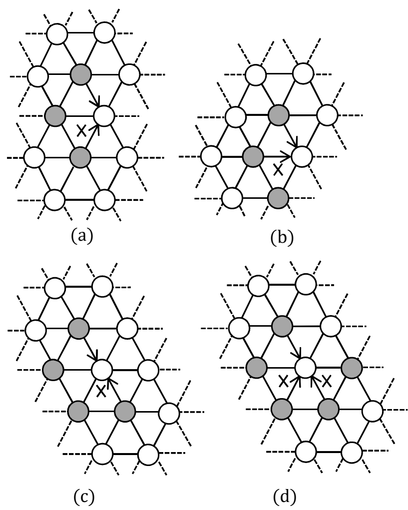

Figure 9: Prohibited behaviors when a robot with one adjacent robot node SW moves to SE.

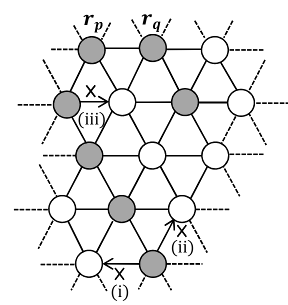

Figure 10: A configuration such that only r_p or r_q can leave the current node ((i): by Lemma 2, (ii): by Fig. 9 (a), (iii): by Fig. 9 (b)).

In the configuration, by Proposition 1 and the above discussion, no robot can leave the current node and robots cannot achieve gathering, which is a contradiction. Thus, we have the following lemma.

Lemma 2. *A robot with one adjacent robot node NW cannot move to node W.*

Case 2: a robot with one adjacent robot node SW moves to SE. In this case, as shown in Fig. 9, the following four robot behaviors are not allowed: (a) a robot with one adjacent robot node NW moves to NE, (b) a robot with two adjacent robot nodes NE and SE moves to E, (c) a robot with one adjacent robot node W moves to NW, and (d) a robot with two adjacent robot nodes NW and E (resp., W and NE) moves to NE (resp., NW). Then, in a configuration of Fig. 10, only robot r_p with two adjacent robot nodes SW and E can move to SE or robot r_q with two adjacent robot nodes W and SE can move to SW by the above discussion and Lemmas 1 and 2 and Collorary 2. We consider each of the behaviors and show for both the cases that robots cannot achieve gathering from some configuration.

Case 2-1: robot r_p moves to SE. In this case, clearly a robot with one adjacent robot node NE cannot move to E and a robot with one adjacent robot node W cannot move to SW since a collision occurs. In addition, when considering a configuration of Fig.11 (a), only robot r_i with one adjacent robot node E can leave the current node and it needs to move to SE by the previous discussions. Now, we consider the configuration of Fig. 12 (a). In the figure, robots r_1, r_3, and r_5 move to SE and the other robots must stay at the current nodes. Then, the system reaches the configuration of Fig. 12 (b). In the configuration, robots r_0, r_2, r_4, and r_6 move to SE and the other robots must stay at the current nodes. Then, the system reaches the configuration of Fig. 12 (a). Thus, robots

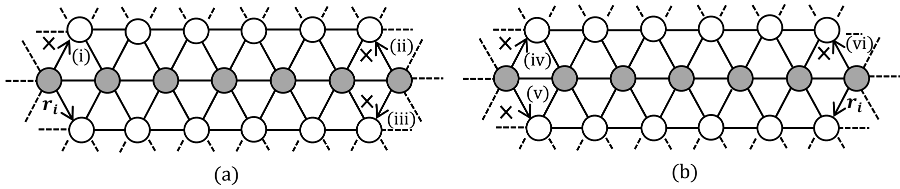

Figure 11: (a): An example such that only a robot r_i with one adjacent robot node E can leave the current node in Case 2-1 ((i): by Fig. 6 (c), (ii): by Fig. 9 (c), (iii): prohibited behavior when a robot with two adjacent robot nodes SW and E moves to SE), (b): An example such that only a robot r_i with one adjacent robot node W can leave the current node in Case 2-2 ((iv): by Fig. 6 (c), (v): prohibited behavior when a robot with two adjacent robot nodes W and SE moves to SW, (vi): by Fig. 9 (c)).

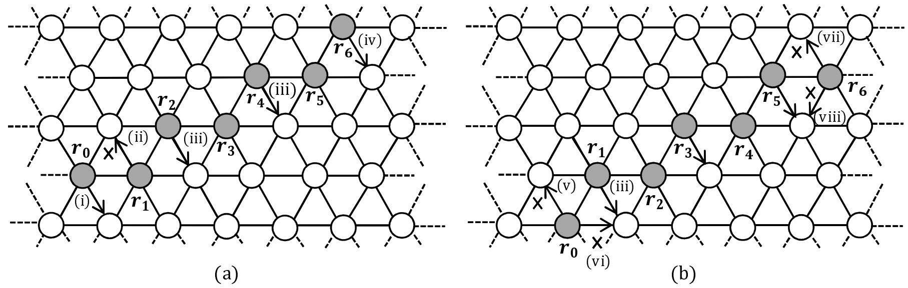

Figure 12: Configurations that robots repeat alternately ((i): by Fig. 11 (a), (ii): by Fig. 9 (d), (iii): assumption of Case 2-1, (iv): assumption of Case 2, (v): by Fig. 6 (a), (vi): prohibited behavior when a robot with two adjacent robot nodes SW and E moves to SE, (vii): by Fig. 9 (c), (viii): prohibited behavior when a robot with two adjacent robot nodes SW and E moves to SE).

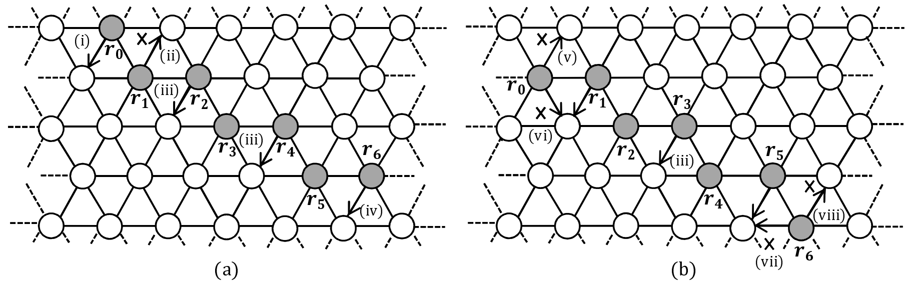

Figure 13: Configurations that robots repeat alternately ((i): assumption of Algorithm $\mathcal{A}$, (ii): by Fig. 9 (d), (iii): assumption of Case 2-2, (iv): by Fig. 11 (b), (v): by Fig. 6 (c), (vi): prohibited behavior when a robot with two adjacent robot nodes W and SE moves to SW, (vii): by Lemma 2, (viii): by Fig. 9 (a)).

repeat configurations of Fig. 12 (a) and (b) forever and they cannot achieve gathering, which is a contradiction.

Case 2-2: robot r_q moves to SW. In this case, clearly a robot with one adjacent robot node E cannot move to SE since a collision occurs. In addition, when considering a configuration of Fig. 11 (b), only robot r_i with one adjacent robot node W can leave the current node and it needs to move to SW by the previous discussions. Now, we consider the configuration of Fig. 13 (a). In the figure, robots r_0, r_2, r_4, and r_6 move to SW and the other robots must stay at the current nodes. Then, the system reaches the configuration of Fig. 13 (b). In the configuration, robots r_1, r_3, and r_5 move to SW and the other robots must stay at the current nodes. Then, the system reaches the

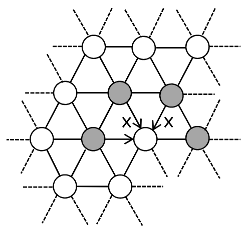

Figure 14: Prohibited behaviors when a robot with one adjacent robot node NE moves to E.

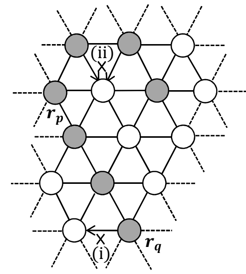

Figure 15: A configuration such that only r_p or r_p can leave the current node ((i): by Lemma 2, (ii): by Fig. 14).

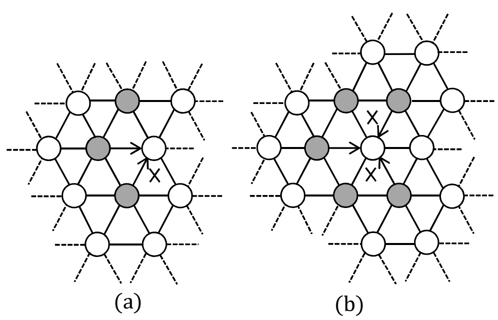

Figure 16: Prohibited behaviors when a robot with two adjacent robot node NE and SE moves to E.

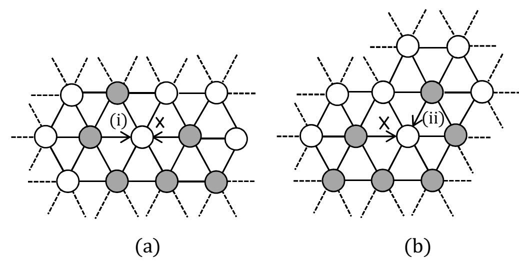

Figure 17: Configurations showing that a robot with two adjacent robot nodes SW and SE must stay at the current node ((i): assumption of Case 3-1, (ii): assumption of Algorithm $\mathcal{A}$).

configuration of Fig. 13 (a). Thus, robots repeat configurations of Fig. 13 (a) and (b) forever and they cannot achieve gathering, which is a contradiction. Thus, we have the following lemma.

Lemma 3. *A robot with one adjacent robot node SW cannot move to node SE.*

Case 3: a robot with one adjacent robot node NE moves to E. In this case, the following two robot behaviors are not allowed (Fig. 14): a robot with two adjacent robot nodes SW and E (resp., W and SE) moves to SE (resp., SW). Then, in a configuration of Fig. 15, it is necessary that at least a robot r_p with two adjacent robot nodes NE and SE moves to E or a robot r_q with one adjacent robot node with NW moves to NE by the previous discussions. We consider each of the behaviors and show for both the cases that robots cannot achieve gathering from some configuration.

Case 3-1: robot r_p moves to E. In this case, the following two robot behaviors are not allowed (Fig. 16): (a) a robot with one adjacent robot node NW moves to NE and (b) a robot with one adjacent robot node W moves to NW or SW. In addition, when considering configurations of Fig. 17, a robot with two adjacent robot node SW and SE must stay at the current node. Then, in a configuration of Fig. 18 only robot r_i with three adjacent robot nodes NW, NE, and SE can leave the current node and it needs to move to E to avoid a collision or an unconnected configuration. In addition, when considering configurations of Fig. 19, a robot r_j with four adjacent robots nodes E, SW, W, and NW must stay at the current node. Hence, in a configuration of Fig. 20, only robot r_i with two adjacent robot nodes NW and NE can leave the current node. Then, in a configuration of Fig. 21, a collision occurs when r_i moves to W. Hence, it needs to move to E. Next, we consider the behavior of robot r_j with two adjacent robot nodes E and SE. When it moves to SW, the configuration becomes unconnected if its southeast robot r_i has two adjacent robot nodes NW and NE, and moves to E (Fig. 22 (a)). In addition, r_j cannot move to NE since a collision occurs in a configuration of Fig. 22 (b). Thus, robot r_j must stay at the current node. Finally, in a configuration of Fig. 23, no robot can leave the current node and robots cannot achieve gathering. Therefore, a

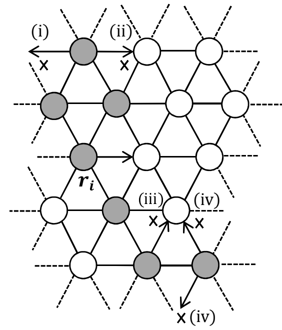

Figure 18: A configuration such that only robot r_i with three adjacent robot nodes NW, NE, and SE can leave the current node ((i): by Fig. 17 (a), (ii): by Fig. 17 (b), (iii): by Fig. 6 (d), (iv): by Fig. 16 (b)).

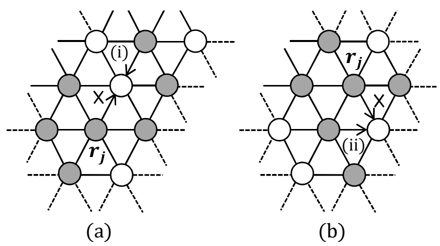

Figure 19: Configurations showing that robot r_j with four adjacent robot nodes E, SW, W, and NW must stay at the current node ((i): assumption of Algorithm $\mathcal{A}$, (ii): by Fig. 18).

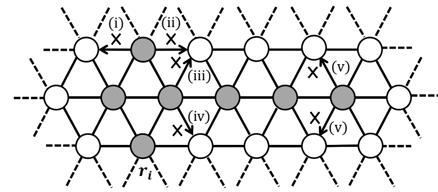

Figure 20: A configuration such that only robot r_i with two adjacent robot nodes NE and NW can leave the current node ((i): by Fig. 17 (a), (ii): by Fig. 17 (b), (iii): by Fig. 19 (a), (iv): by Fig. 19 (b), (v): by Fig. 16 (b)).

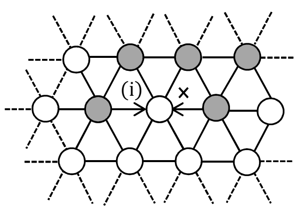

Figure 21: An example showing that a robot with two adjacent robot nodes NE and NW cannot move to W ((i): assumption of Case 3).

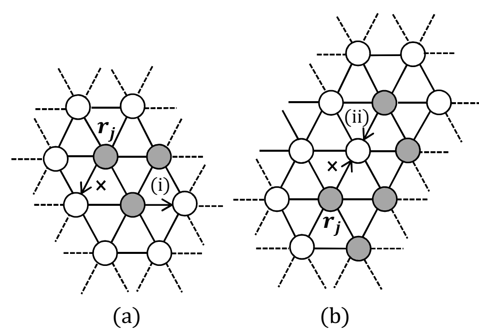

Figure 22: Configurations showing that a robot with two adjacent robot nodes SW and SE must stay at the current node ((i): by Fig. 21, (ii): assumption of Algorithm $\mathcal{A}$).

robot r_p with two adjacent robot nodes NE and SE must stay at the current node. By a similar discussion, we can also show that a robot with three adjacent robot nodes NE, SE, and SW must stay at the current node.

Case 3-2: robot r_q moves to NE. In this case, when considering a configuration of Fig. 24, only robot r_i with two adjacent robot nodes E and NE can leave the current node. However, r_i cannot move to NW since a collision occurs in a configuration of Fig. 25 and it needs to move to SE. Similarly, in a configuration of Fig. 26 only robot r_j with two adjacent robot nodes NW and W can

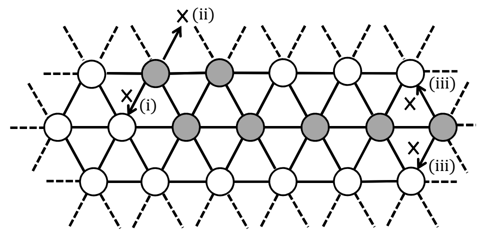

Figure 23: An unsolvable configuration when a robot with two adjacent robot nodes NE and SE moves to E ((i): by Fig. 22 (a), (ii): by Fig. 22 (b), (iii): by Fig. 16 (b)).

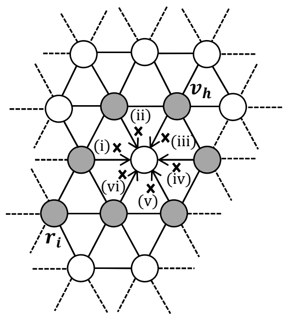

Figure 24: A configuration such that only robot r_i with two adjacent robot nodes NE and E can leave the current node ((i): by Case 3-1, (ii), (iii): by Fig. 14, (iv)-(vi): behaviors that may cause a collision if a robot at v_h has one adjacent robot node SE and moves to SW by the hypothesis of Algorithm $\mathcal{A}$).

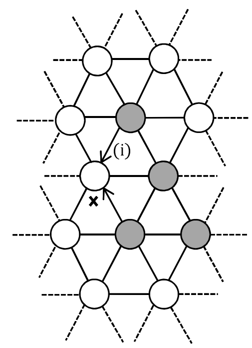

Figure 25: An example showing that a robot with two adjacent robot nodes NE and E cannot move to NW ((i): assumption of Algorithm $\mathcal{A}$).

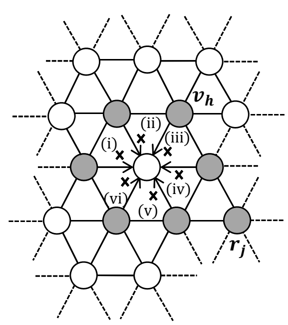

Figure 26: A configuration such that only robot r_i with two adjacent robot nodes NW and W can leave the current node ((i): by Case 3-1, (ii), (iii): by Fig. 14, (iv)-(vi): behaviors that may cause a collision if a robot at v_h has one adjacent robot node SE and moves to SW by the hypothesis of Algorithm $\mathcal{A}$).

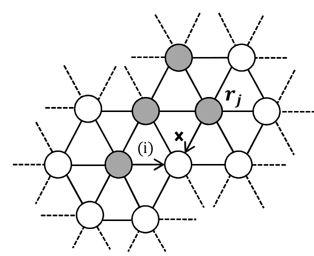

Figure 27: An example showing that a robot with two adjacent robot nodes NW and W cannot move to SW ((i): assumption of Case 3).

leave the current node. However, r_j cannot move to SW since a collision occurs in a configuration of Fig. 27 and it needs to move to NE. Now, we consider the configuration of Fig. 28. In the figure,

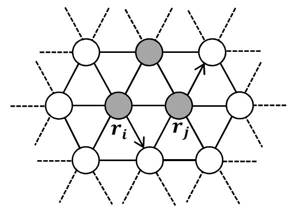

Figure 28: A configuration that becomes unconnected after robots r_i and r_j move.

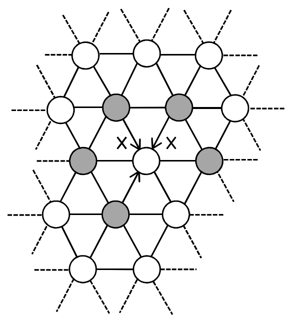

Figure 29: Prohibited behaviors when a robot with one adjacent robot node NW moves to NE.

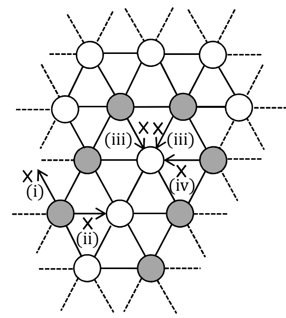

Figure 30: A configuration such that no robot can leave the current node when a robot with one adjacent robot node NW moves to NE ((i): by Fig. 6 (a), (ii): by Lemma 4, (iii): by Fig. 29, (iv): by Fig. 6 (b)).

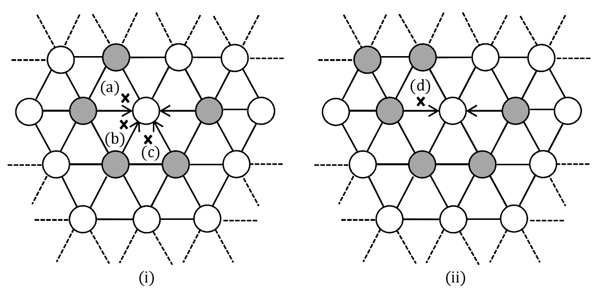

Figure 31: Prohibited behaviors when a robot with one adjacent robot node SW moves to W.

robot r_i moves to SE and robot r_j moves to NE by the above discussion. Then, the configuration becomes unconnected and robots cannot achieve gathering. Thus, a robot r_q with one adjacent robot node NW must stay at the current node. Therefore, robots cannot solve the problem from the configuration of Fig. 15 and we have the following lemma.

Lemma 4. *A robot with one adjacent robot node NE cannot move to node E.*

Case 4: a robot with one adjacent robot node NW moves to NE. In this case, the following two robot behaviors are not allowed (Fig. 29): a robot with two adjacent robot nodes SW and E (resp., W and SE) moves to SE (resp., SW). Then, in a configuration of Fig. 30, no robot can leave the current node and robots cannot achieve gathering. Thus, we have the following lemma.

Lemma 5. *A robot with one adjacent robot node NW cannot move to node NE.*

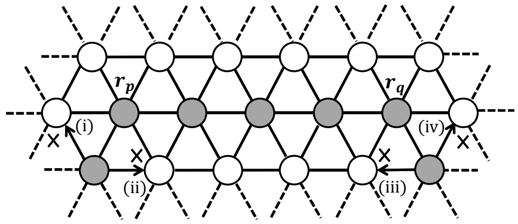

Figure 32: A configuration such that only robot r_p with two adjacent robot nodes SW and E or r_q with two adjacent robot nodes W and SE can leave the current node ((i): by Fig 6 (a), (ii): by Lemma 4, (iii): by Lemma 2, and (iv): by Lemma 5).

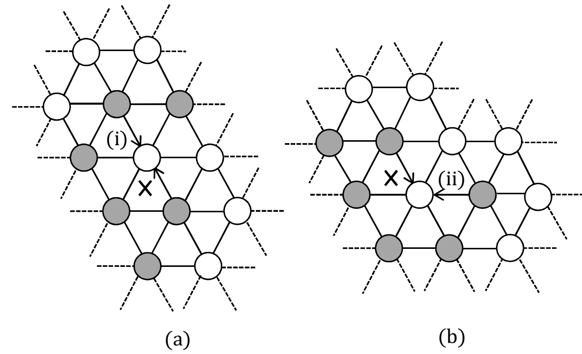

Figure 33: Examples showing that a robot with two adjacent robot nodes W and SW cannot leave the current node ((i): assumption of Case 5-1, (ii): assumption of Case 5).

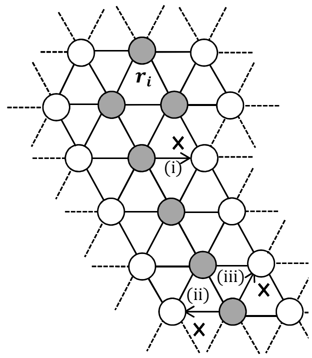

Figure 34: A configuration such that only robot r_i with two adjacent robot nodes SE and SW can leave the current node ((i): by Fig 31 (d), (ii): by Lemma 2, (iii): by Lemma 5).

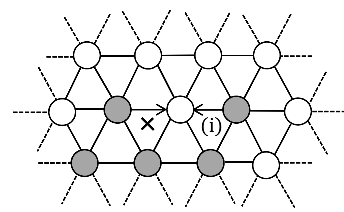

Figure 35: An example showing that a robot with two adjacent robot nodes SE and SW cannot move to E. ((i): assumption of Case 5).

Case 5: a robot with one adjacent robot node SW moves to W. In this case, the following four robot behaviors are not allowed (Fig. 31): (a): a robot with two adjacent robot nodes NE and SE moves to E, (b): a robot with two adjacent robot nodes NW and E moves to NE, (c): a robot with two adjacent robot nodes W and NE moves to NW, and (d): a robot with three adjacent robot nodes NE, NW, and SE moves to E. Then, in a configuration of Fig. 32, it is necessary that robot r_p with two adjacent robot nodes SW and E moves to SE or robot r_q with two adjacent robot nodes W and SE moves to SW. We consider each of the behaviors and show for both the cases that robot cannot achieve gathering from some configuration.

Case 5-1: robot r_p moves to SE. In this case, a robot with two adjacent robot nodes SW and W cannot leave the current node because otherwise a collision may occur (Fig. 33). Next, when considering a configuration of Fig. 34, only robot r_i with two adjacent robot nodes SE and SW can leave the current node. However, in a configuration of Fig. 35, r_i cannot move to E since a collision occurs. Hence, r_i needs to move to W. Then, a robot r_j with two adjacent robot nodes E and NE cannot move to NW or SE because a collision occurs or the configuration becomes unconnected (Fig. 36). Finally, we consider the configuration of Fig. 37. In the figure, no robot can leave the current node and they cannot achieve gathering. Thus, a robot r_p with two adjacent robot nodes SW and E cannot move to SE and it must stay at the current node.

Case 5-2: robot r_q moves to SW. In this case, a robot with one adjacent robot node E cannot

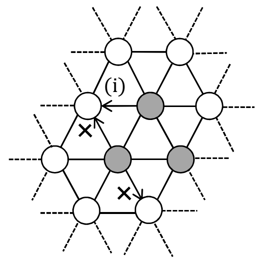

Figure 36: An example showing that a robot with two adjacent robot nodes E and NE cannot leave the current node ((i): by Fig. 35).

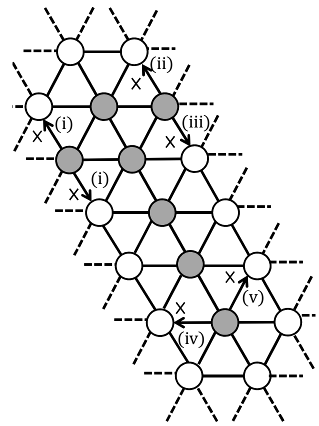

Figure 37: An unsolvable configuration when a robot with two adjacent robot nodes SW and E moves to SE ((i): by Fig. 36, (ii): by Fig. 33 (a), (iii): by Fig. 33 (b), (iv): by Lemma 2, (v): by Lemma 5).

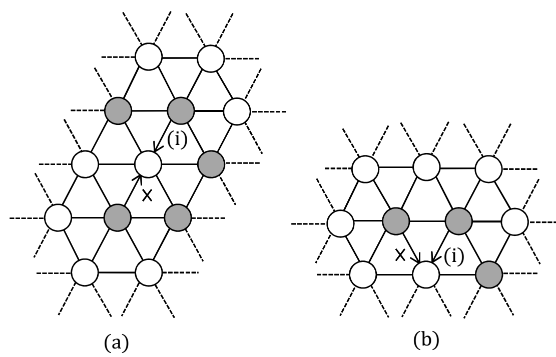

Figure 38: Examples showing that a robot with two adjacent robot nodes E cannot leave the current node ((i): assumption of Case 5-2).

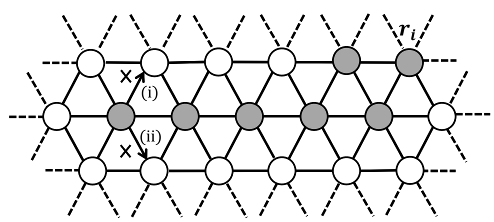

Figure 39: A configuration such that only robot r_i with two adjacent robot nodes W and SW can leave the current node ((i): by Fig. 38 (a), (ii): by Fig. 38 (b)).

leave the current node because otherwise a collision may occur (Fig. 38). Then, when considering a configuration of Fig. 39, only robot r_i with two adjacent robot nodes SW and W can leave the current node. However, r_i cannot move to SE since a collision occurs in a configuration of Fig. 40. Hence, r_i needs to move to NW. Then, robot r_j with two adjacent robot nodes E and SE cannot leave the current node because a collision occurs or the configuration becomes unconnected (Fig. 41).

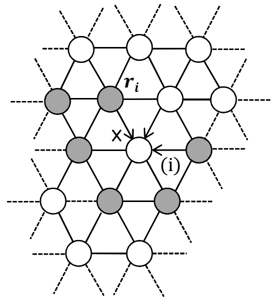

Figure 40: An example showing that a robot with two adjacent robot nodes SW and W cannot move to SE ((i): assumption of Case 5).

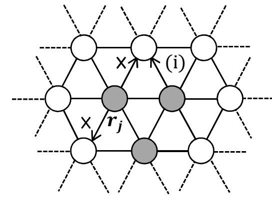

Figure 41: An example showing that a robot with two adjacent robot nodes E and SE cannot leave the current node ((i): by Fig. 40).

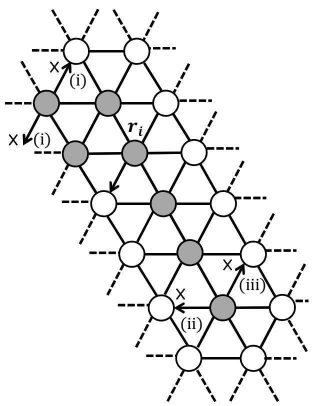

Figure 42: A configuration such that only robot r_i with three adjacent robot nodes SE, W, and NW can leave the current node ((i): by Fig. 41, (ii): by Lemma 2, (iii): by Lemma 5).

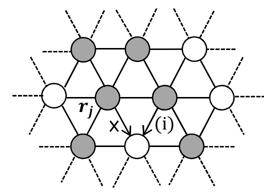

Figure 43: An example showing that a robot r_j with four adjacent robot nodes E, SW, NW, and NE cannot move to SE ((i): by Fig. 42).

Next, we consider the configuration of Fig. 42. In the figure, only robot r_i with three adjacent robot nodes SE, W, and NW can leave the current node and it needs to move to SW by the previous discussions. Then, a robot r_j with four adjacent robot nodes E, SW, NW, and NE cannot leave the current node since a collision occurs in a configuration of Fig. 43. Thus, when considering the configuration of Fig. 44, only robot r_i with two adjacent robot nodes W and NW can leave the current node. However, r_i cannot move to NE since a collision occurs in a configuration of Fig. 45. Thus, r_i needs to move to SW. Then, a robot with one adjacent robot node W cannot leave the current node because otherwise a collision occurs in configurations of Fig. 46. Finally, when considering the configuration of Fig. 47, no robot can leave the current node and robots cannot achieve gathering, which is a contradiction. Thus, a robot r_q with two adjacent robot nodes W and SE cannot move to SW. Therefore, we have the following lemma.

Lemma 6. *A robot with one adjacent robot node SW cannot move to node W.*

Thus, by Proposition 1-(a), Lemmas 1, 3, 4, and 5, robots cannot achieve gathering from a configuration of Fig. 4 (b) and we have the theorem. □

Remark. We showed impossibility of gathering for seven mobile robots which requires that robots form a hexagon with radius 1. This result also holds for gathering of the specific number of robots such that they form a hexagon with radius of a positive integer after gathering.

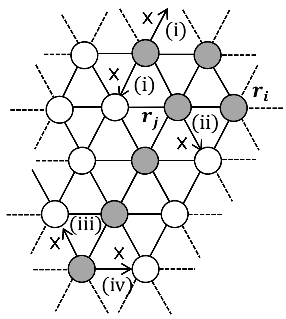

Figure 44: A configuration such that only robot r_i with two adjacent robot nodes W and NW can leave the current node ((i): by Fig. 41, (ii): by Fig. 43, (iii): by Fig. 6 (a) (iv): by Lemma 4).

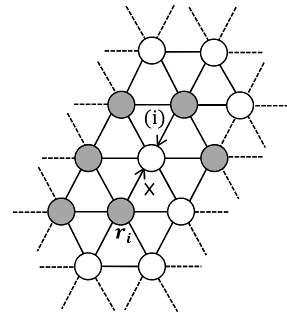

Figure 45: An example showing that a robot r_i with two adjacent robot nodes W and NW cannot move to NE ((i): assumption of Case 5-2).

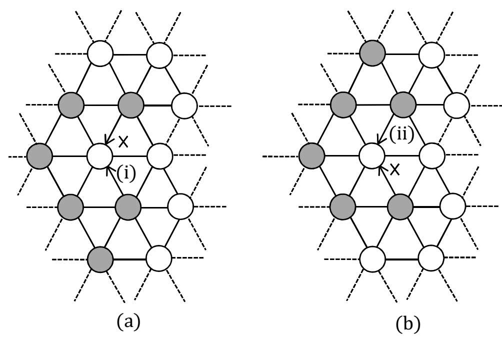

Figure 46: Examples showing that a robot with one adjacent robot node W cannot leave the current node ((i): by Fig. 40, (ii): by Fig. 45).

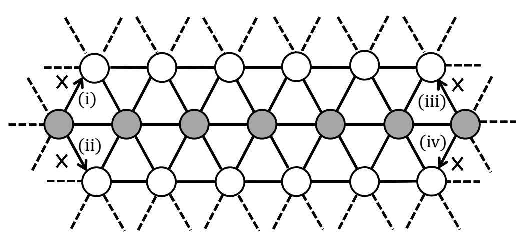

Figure 47: An unsolvable configuration when a robot with two adjacent robot nodes W and SE moves to SW ((i): by Fig. 38 (a), (ii): by Fig. 38 (b), (iii): by Fig. 46 (b), (iv): by Fig. 46 (a)).

4 Robots with visibility range 2

In this section, for robots with visibility range 2, we propose a collision-free algorithm to solve the gathering problem from any connected initial configuration.

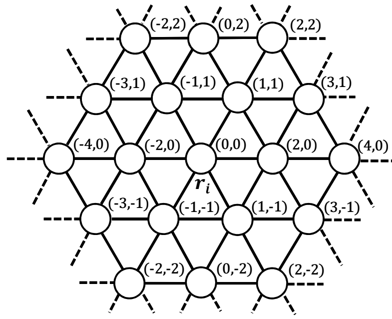

Figure 48: Assignment of labels.

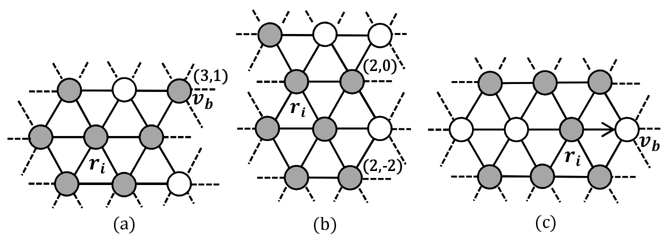

Figure 49: Examples of how to determine the base nodes ((a): node v_b is the base node, (b): r_i does not determine the base node, (c): r_i determines v_b as the base node and moves there).

4.1 Proposed algorithm

The basic idea is that each robot firstly determines the *base node* that is the rightmost robot node within its visibility range and then it moves toward the base node to achieve gathering. First, we explain how to determine the base (or rightmost) robot. For explanation, in the following we assume that each robot r_i recognizes that it is located at an origin and it assigns labels to each node within its visibility range like Fig. 48. In the figure, the first (resp., second) element of each label is called the x-*element* (resp., y-*element*)[2]. Then, r_i determines the robot node with the largest x-element as the base node (possibly the robot node where r_i itself stays). If several robot nodes have the largest x-element, r_i does not determine the base node at that time and waits at the current node until the configuration changes. As exceptions, if node (4,0) is an empty node and nodes (3,1) and (3,-1) are robot nodes, r_i determines node (4,0) as the base node to avoid the configuration such that no robot determines a base node and each robot waits at the current node. In addition, if robot nodes (1,1) and (1,-1) have the largest x-element among all the labels of robot nodes within r_i's visibility range, and r_i moves to node (2,0) so that it becomes a base. Examples are given in Fig. 49.

Next, we explain how to achieve gathering based on the base node. Robots consider the base node as the rightmost node of a gathering-achieved configuration and they basically move east on a triangular grid with avoiding a collision and an unconnected configuration. Concretely, if the label of the base node from robot r_i is (2,-2), (3,-1), (4,0), (3,1), or (2,2), it moves to one of adjacent nodes as indicated in Fig. 50 (a) using ordinal numbers in Fig. 50 (b). That is, among the candidate nodes that r_i may visit in the next cycle, r_i moves to the empty adjacent node with the smallest ordinal number. If several robots try to move to the same node v_j, the robot staying at the node with the largest ordinal number moves to v_j. If all the candidate nodes are robot nodes, r_i stays at the current node. For example, in Fig. 51, robots r_i and r_j consider the common node v_b as the base node, r_i (resp., r_j) has two candidate nodes v_n and v_ℓ (resp., v_m and v_ℓ) to visit, v_n (resp., v_m) is a robot node, and hence it tries to visit v_ℓ (resp., v_ℓ). In this case, since the ordinal number 4 of the node where r_i stays is larger than the ordinal number 3 of the node where r_j stays, r_i moves

[2]Labels are assigned for explanation and they are a little different from the coordinate system. For example, the difference between labels (0,0) and (2,0) is 2 but the distance between node (0,0) and node (2,0) is 1.

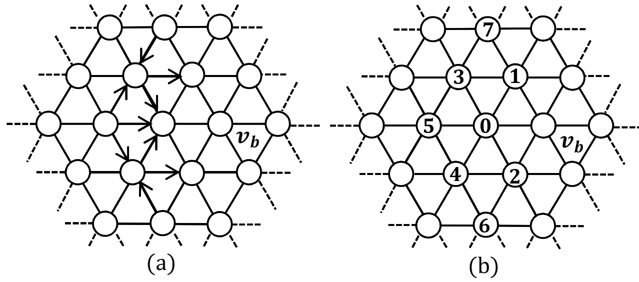

Figure 50: Movement rules ((a): candidate nodes to visit, (b): ordinal numbers).

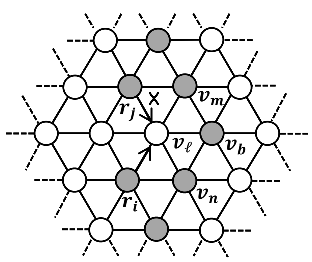

Figure 51: An example to avoid a collision using ordinal numbers.

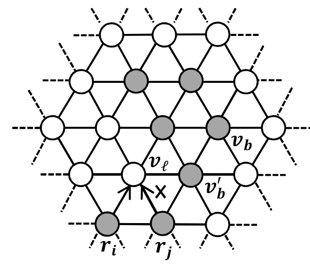

Figure 52: An example to avoid a collision using x-elements.

to v_ℓ and r_j stays at the current node. If two robots consider the common node as the base node like the above example, they can share the common ordinal numbers and can avoid a collision or an unconnected configuration. However, it is possible that some two robots consider different robot nodes as their base nodes due to their limited visibility range, which may cause a collision or an unconnected configuration. For example, in Fig. 52, robot r_i considers v_b' as the base node but r_j considers v_b as the base node, and they try to move to the same node v_ℓ according to the movement rule. In this case, the robot with the smaller x-element of the node label moves to the node and the other robot stays at the current node. Hence, in Fig. 52, r_i moves to v_ℓ and r_j stays at the current node. Moreover, only with the movement rule in Fig. 50, no robot leaves the current node in the configuration in Fig. 53. In this case, as a special behavior, if the label of the base node from robot r_i is (3,1), nodes (1,1), (2,0), and (1,-1) are robot nodes, and node (-1,1) is an empty node, r_i moves to the northwest adjacent node (-1,1) so that robot r_j staying at r_i's southeast adjacent node (1,-1) can move to the node where r_i is currently staying. Similarly, as a special behavior, if the label of the base node from robot r_i is (3,-1), nodes (1,-1), (2,0), and (1,1) are robot nodes, and node (-1,-1) is an empty node, r_i moves to the southwest adjacent node (-1,-1) so that robot r_j staying at r_i's southeast adjacent node (1,1) can move to the node where r_i is currently staying (we omit the figure). When robots reach a configuration such that no robot leaves the current node, the configuration is one solution of the gathering problem.

An example of the algorithm execution is given in Fig. 54. From (a) to (b), for r_2, since its northeast adjacent robot node (1,1) and southeast adjacent robot node (1,-1) have the largest x-element among robot nodes that r_2 observes, it moves to east adjacent node v_b. From (b) to (c), robots r_0 and r_3 consider v_b as the common base node and they try to move to the empty node v_k with the smallest ordinal number among candidate empty nodes. In this case, since the ordinal number of the node where r_3 stays is larger than that of the node where r_0 stays, r_3 moves to v_k and r_0 stays at the current node. From (c) to (d), r_5 (resp., r_6) considers v_b' (resp., v_b) as the base node and they try to move to node v_ℓ. In this case, since the x-element of the node that where r_5 stays is smaller than that of the node where r_6 stays, r_5 moves to v_ℓ and r_6 stays at the current node. From (d) to (e), as a special behavior, robot r_5 moves to the northwest adjacent robot node so that r_6 can move to the node where r_5 is currently staying. From (e) to (f), robot r_6 considers

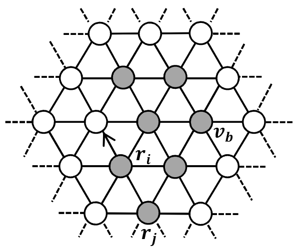

Figure 53: An example to avoid a standstill.

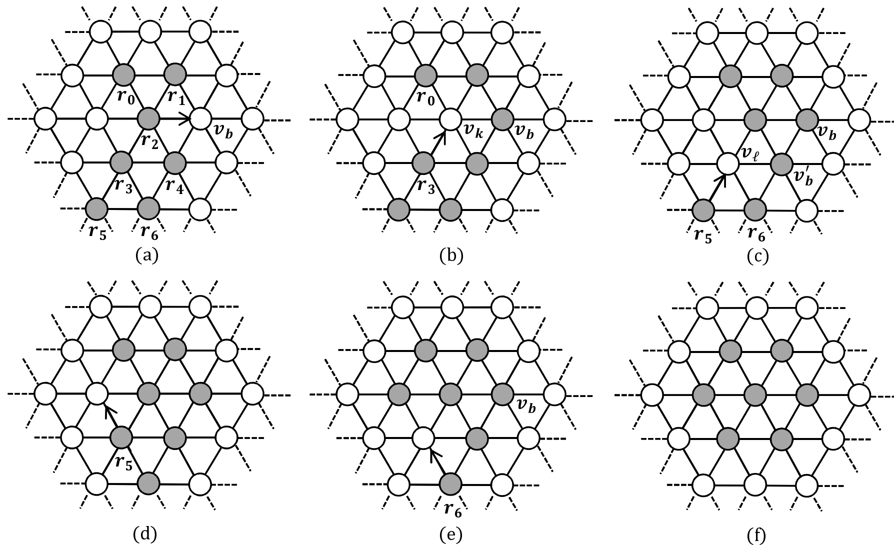

Figure 54: An example of the algorithm execution.

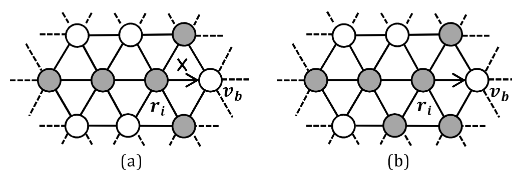

Figure 55: Behavior of robot r_i when the label of the base node v_b is (2,0) but the node is an empty node.

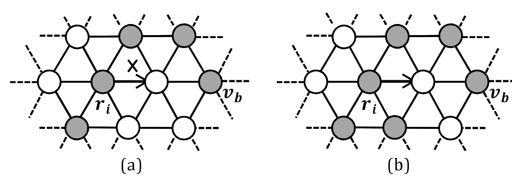

Figure 56: Behavior of robot r_i when the label of the base node v_b is (4,0).

v_b as the base node and it moves to the northwest adjacent node. Then, robots achieve gathering.

The pseudocode of the proposed algorithm is described in Algorithm 1. In the following, we explain several robot behaviors that avoid a collision or an unconnected configuration. The behavior of robot r_i for the case that, the label of the base node is (2,0) but the node is an empty node, is described in lines 1 – 3. In this case, r_i tries to move to node (2,0). However, if r_i's west adjacent node (-2,0) is a robot node and r_i moves to the base node (2,0), the configuration may become unconnected (Fig. 55 (a)). Hence, in this case r_i moves to node (2,0) when r_i's northwest or southwest adjacent node is also a robot node (Fig. 55 (b)).

The behavior of robot r_i for the case that the label of the base node is (4,0) is described in lines 5 – 9. In this case, if node (2,0) is an empty node, r_i tries to move to the node. However, if r_i's southwest adjacent node (-1,-1) is a robot node and r_i moves to node (2,0), the configuration may become unconnected (Fig. 56 (a)). Hence, in this case r_i moves to node (2,0) when its southeast adjacent node (1,-1) is also a robot node (Fig. 56 (b)).

The behavior of robot r_i for the case that the label of the base node is (3,-1) is described in lines 11 – 15. In this case, if nodes (2,0), (1,-1) and (1,1) are robot nodes and node (-1,-1) is an

Algorithm 1 Proposed algorithm

```
if (node (2,0) is an empty node) ∧ (nodes (1,1) and (1,-1) are robot nodes) ∧ (the other robot nodes have x-elements of the labels at most 0) then
    /*The base node is (2,0) but it is an empty node*/
    if (node (-2,0) is an empty node) ∨ ((node (-2,0) is a robot node) ∧ (node (-1,1) or (-1,-1) is a robot node)) then move to the east adjacent node (2,0)

else if (node label of the base node is (4,0)) ∨ ((node (4,0) is an empty node) ∧ (nodes (3,1) and (3,-1) are robot nodes)) then
    /*The base node is (4,0)*/
    if (node (2,0) is an empty node) ∧ ((nodes (-1,1), (-2,0), and (-1,-1) are empty nodes) ∨ (node (1,-1) is a robot node and nodes (-2,0) and (-1,1) are empty nodes) ∨ (node (1,1) is a robot node and nodes (-2,0) and (-1,-1) are empty nodes) ∨ (nodes (1,-1), (-1,-1), and (-2,0) are robot nodes and node (-1,1) is an empty node) ∨ (nodes (-2,0), (-1,1) and (1,1) are robot nodes and node (-1,-1) is an empty node)) then move to the east adjacent node (2,0)
    else if (node (2,0) is a robot node) ∧ (node (1,1) is an empty node) ∧ (nodes (-2,0) and (-1,1) are empty nodes) ∧ ((nodes (-1,-1) and (2,2) are empty nodes) ∨ (nodes (2,2), (3,1), (3,-1), and (-2,-2) are robot nodes)) then move to the northeast robot node (1,1)
    else if (nodes (2,0) and (1,1) are robot nodes) ∧ (nodes (1,-1) is an empty node) ∧ (nodes (-1,-1) (-2,0), (-1,1), and (2,-2) are empty nodes) ∧ ((node (1,1) is a robot node) ∨ (node (2,2) is a robot node)) then move to the southeast adjacent node (1,-1)

else if node label of the base node is (3,-1) then
    /*The base node is (3,-1)*/
    if (node (1,-1) is an empty node) ∧ (nodes (-1,-1) and (0,-2) are empty nodes) ∧ ((nodes (-2,0) and (-1,1) are empty nodes) ∨ (nodes (-1,1) and (1,1) are robot nodes and node (0,2) is an empty node)) then move to the southeast adjacent node (1,-1)
    else if (node (1,-1) is a robot node) ∧ (node (2,0) is an empty node) ∧ (node (-1,1) is an empty node) ∧ ((node (-2,0) is an empty node) ∨ (nodes (-2,0) and (-1,-1) are robot nodes)) then move to the east adjacent node (2,0)
    else if (nodes (1,-1) and (2,0) are robot nodes) ∧ (node (1,1) is a robot node) ∧ (node (-1,-1) is an empty node) ∧ (nodes (-2,0) and (-2,-2) are empty node) then move to the southwest node (-1,-1)

else if node label of the base node is (2,-2) then
    /*The base node is (2,-2)*/
    if (node (-1,-1) is an empty node) ∧ (nodes (-2,0), (-3,-1), and (-1,1) are empty nodes) then move to the southwest adjacent node (-1,-1)

else if node label of the base node is (3,1) then
    /*The base node is (3,1)*/
    if (node (1,1) is an empty node) ∧ ((nodes (-1,1), (-2,0), (-1,-1) are empty nodes) ∨ (nodes (1,-1) and (-1,-1) are robot nodes and nodes (0,-2) and (-1,1) are empty node)) then move to the northeast adjacent node (1,1)
    else if (node (1,1) is a robot node) ∧ (node (2,0) is an empty node) ∧ ((nodes (-2,0) and (-1,-1) are empty nodes) ∨ (node (-1,-1) is an empty node and nodes (-2,0) and (-1,1) are robot nodes)) then move to the east adjacent node (2,0)
    else if (nodes (1,1) and (2,0) are robot nodes) ∧ (node (1,-1) is a robot node) ∧ (node (1,-1) is an empty node) ∧ (nodes (-2,0), and (-2,2) are empty nodes) then move to the northwest adjacent node (-1,1)

else if node label of the base node is (2,2) then
    /*The base node is (2,2)*/
    if (node (-1,1) is an empty node) ∧ (nodes (-3,1), (-2,0), and (-1,-1) are empty nodes) then move to its northwest adjacent node (-1,1)

else if (node label of the base node is (0,0) or (2,0) or (1,-1) or (1,1)) ∨ (there is no base node) then
    /*Robot r_i is close to the base node and it does not need to leave the current node*/
    stay at the current node
end if
```

empty node, as a special behavior, r_i tries to move to its southwest adjacent node (-1,-1) so that the robot staying at node (1,1) could move to node (0,0) where r_i is currently staying. However, due to the limited visibility range, it is possible that r_i and some robot r_j consider different nodes as base nodes, r_j staying at node (-2,0) or (-2,-2) tries to move to node (-1,-1), and a collision occurs

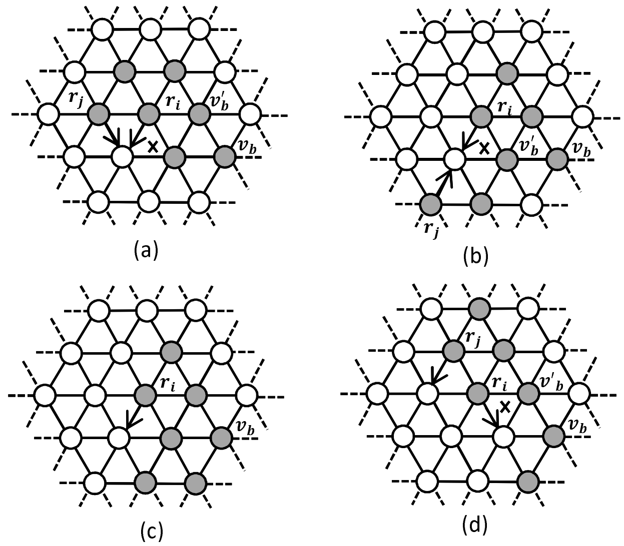

Figure 57: Behavior of robot r_i when the label of the base node v_b is (3,-1) (v'_b: the base node for robot r_j).

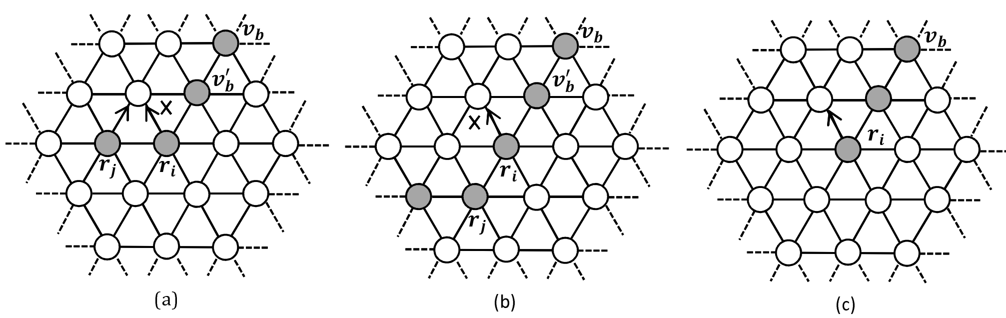

Figure 58: Behavior of robot r_i when the label of the base node v_b is (2,2) (v'_b: the base node for robot r_j).

(Fig. 57 (a), (b)). Hence, in this case r_i moves to node (-1,-1) when nodes (-2,0), (-2,-2), and (-1,1) are empty nodes (Fig. 57 (c)). In addition, if node (3,-1) is a base node and node (1,-1) is an empty node, r_i tries to move to the node. Then, it is possible that r_i and some robot r_j consider different nodes are base nodes, r_j staying at node (-1,1) tries to move to node (-2,0), and the configuration become unconnected (Fig. 57 (d)). Hence, in this case r_i moves to node (1,-1) when node (0, 2) is an empty node.

The behavior of robot r_i for the case that the label of the base node is (2,2) is described in lines 27 – 29. In this case, if node (1,1) is a robot node and node (-1,1) is an empty node, it tries to move to node (-1,1). However, due to the limited visibility range, it is possible that r_i and some robot r_j consider different nodes are base nodes, r_j staying at node (-2,0) tries to move to node (-1,1), and a collision occurs (Fig. 58 (a)), or r_j staying at node (-1,-1) does not leave the current node and the configuration becomes unconnected (Fig. 58 (b)). Hence, in this case r_i moves to node (-1,-1) when nodes (-2,0) and (-1,-1) are empty nodes (Fig. 58 (c)).

Although there still exist several robot behaviors that avoid a collision or an unconnected configuration, we omit the detail.

4.2 Correctness

The correctness of the proposed algorithm has been evaluated by computer simulations. By the simulations, we confirmed that robots which execute the proposed algorithm can achieve gathering from all possible connected initial configurations (3652 patterns in total) in the fully synchronous (FSYNC) model. Thus, we have the following theorem.

Theorem 2. *For robots with visibility range 2, the proposed algorithm solves the gathering problem from any connected initial configuration in the FSYNC model.*

5 Conclusion

In this paper, we considered the gathering problem of seven autonomous mobile robots on triangular grid graphs. First, for robots with visibility range 1, we showed that no collision-free algorithm exists for the gathering problem. Next, for robots with visibility range 2, we proposed a collision-free algorithm to solve the problem from any connected initial configuration. This algorithm is optimal in terms of visibility range.

There are four possible future works as follows. First, we will complete a theoretical proof of correctness for the proposed algorithm in Section 4. Second, we will consider the relaxed version of connected initial configurations such that the visibility relationship among robots constitutes one connected graph. Third, we will consider gathering for different number of robots. Lastly, we consider other problems such as the pattern formation problem for autonomous mobile robots on triangular grids.

Acknowledgement

This work was partially supported by JSPS KAKENHI Grant Number 18K18029, 18K18031, 19K11823, 20H04140, 20KK0232, and 21K17706; the Hibi Science Foundation; and Foundation of Public Interest of Tatematsu.

References

[1] I. Suzuki and M. Yamashita. Distributed anonymous mobile robots: Formation of geometric patterns. SIAM Journal on Computing, 28(4):1347–1363, 1999.

[2] P. Courtieu, L. Rieg, S. Tixeuil, and X. Urbain. Impossibility of gathering, a certification. Information Processing Letters, 115(3):447–452, 2015.

[3] Y. Dieudonné and F. Petit. Self-stabilizing gathering with strong multiplicity detection. Theoretical Computer Science, 428:47–57, 2012.

[4] G. D'Angelo, G. Di Stefano, R. Klasing, and A. Navarra. Gathering of robots on anonymous grids and trees without multiplicity detection. Theoretical Computer Science, 610:158–168, 2016.

[5] J. Castenow, M. Fischer, J. Harbig, D. Jung, and FM. auf der Heide. Gathering anonymous, oblivious robots on a grid. Theoretical Computer Science, 815:289–309, 2020.

[6] R. Klasing, A. Kosowski, and A. Navarra. Taking advantage of symmetries: Gathering of many asynchronous oblivious robots on a ring. Theoretical Computer Science, 411(34-36):3235–3246, 2010.

[7] R. Klasing, E. Markou, and A. Pelc. Gathering asynchronous oblivious mobile robots in a ring. Theoretical Computer Science, 390(1):27–39, 2008.

[8] G. D'Angelo, G. Di Stefano, and A. Navarra. Gathering on rings under the look–compute–move model. Distributed Computing, 27(4):255–285, 2014.

[9] G. Di Stefano and A. Navarra. Optimal gathering of oblivious robots in anonymous graphs and its application on trees and rings. Distributed Computing, 30(2):75–86, 2017.

[10] T. Izumi, T. Izumi, S. Kamei, and F. Ooshita. Time-optimal gathering algorithm of mobile robots with local weak multiplicity detection in rings. IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences, 96(6):1072–1080, 2013.

[11] S. Kamei, A. Lamani, F. Ooshita, and S. Tixeuil. Asynchronous mobile robot gathering from symmetric configurations without global multiplicity detection. Proceedings of the 18th International Colloquium on Structural Information and Communication Complexity (SIROCCO), pages 150–161, 2011.

[12] S. Kamei, A. Lamani, F. Ooshita, and S. Tixeuil. Gathering an even number of robots in an odd ring without global multiplicity detection. International Symposium on Mathematical Foundations of Computer Science (MFCS), pages 542–553, 2012.

[13] J. Czyzowicz, L. Gasieniec, and A. Pelc. Gathering few fat mobile robots in the plane. Theoretical Computer Science, 410(6-7):481–499, 2009.

[14] C. Agathangelou, C. Georgiou, and M. Mavronicolas. A distributed algorithm for gathering many fat mobile robots in the plane. Proceedings of the 2013 ACM symposium on Principles of distributed computing (PODC), pages 250–259, 2013.

[15] Y. Ito, Y. Katayama, and K. Wada. A gathering problem for nobile fat robots in a grid without the global coordinate system. Technical Report of IEICE (COMP2014-10), 114:53–59, 2014.

[16] Z. Derakhshandeh, S. Dolev, R. Gmyr, AW. Richa, C. Scheideler, and T. Strothmann. Brief announcement: amoebot–a new model for programmable matter. In Proceedings of the 26th ACM symposium on Parallelism in algorithms and architectures (SPAA), pages 220–222. ACM, 2014.

[17] J. J Daymude, R. Gmyr, A. W Richa, C. Scheideler, and T. Strothmann. Improved leader election for self-organizing programmable matter. International Symposium on Algorithms and Experiments for Sensor Systems, Wireless Networks and Distributed Robotics (ALGOSENSORS), pages 127–140, 2017.

[18] S. Cannon, JJ. Daymude, D. Randall, and AW. Richa. A markov chain algorithm for compression in self-organizing particle systems. Proceedings of the 2016 ACM Symposium on Principles of Distributed Computing (PODC), pages 279–288, 2016.

[19] G. A. Di Luna, P. Flocchini, N. Santoro, G. Viglietta, and Y. Yamauchi. Shape formation by programmable particles. Distributed Computing, 33(1):69–101, 2020.

[20] Z. Derakhshandeh, R. Gmyr, AW. Richa, C. Scheideler, and T. Strothmann. Universal shape formation for programmable matter. Proceedings of the 2016 ACM Symposium on Principles of Distributed Computing (PODC), pages 289–299, 2016.