Gathering Multiple Robots in a Ring and an Infinite Grid

Durjoy Dutta, Tandrima Dey, and Sruti Gan Chaudhuri(✉)

Department of Information Technology, Jadavpur University, Kolkata, India
srutiganc@it.jusl.ac.in

Abstract. Gathering can be coined as one of the primary interaction parameters in systems of autonomous mobile agents or sensors, known as robots. These robots are identical and placed in the nodes of an unlabeled graph. They operate in wait-look-compute-move cycles. In one cycle, first the sensors of the robots are activated independent of each other (wait). Then a robot takes a snapshot of the current configuration (look), makes a decision to stay idle or to move to one of its adjacent nodes (compute), and in the latter case makes an instantaneous move to this neighbor (move). Then the robot again goes back to its initial phase (wait). Cycles are performed asynchronously for each robot. The robots are oblivious, i.e., they do not use any computed data from the previous cycle. The robots do not agree on a common coordinate system. They cannot differentiate between a node having single robot and a node having multiple robots, i.e., multiplicity of a node. The robots are not able to see all the nodes of the graph. They do not know the total number of robots in the system. In this paper, we have developed two algorithms to gather these robots at a single node (not known beforehand) of a Ring Graph and an infinite Grid, in finite time. To the best of our knowledge, this is one of the first reported results on gathering multiple robots under limited visibility in an infinite grid and a ring.

Keywords: Gathering · Limited visibility · Ring · Infinite grid · Asynchronous · Oblivious · Mobile robots

1 Introduction

The swarm robots are a system of multiple autonomous identical mobile robots work in collaboration to execute a given task. Though large in numbers, the cost of this system of robots is comparatively lower than a traditional big robot. Operated by simple hardware and software, these robots are easy to deploy and maintain even in very harsh environment. These robots are used to locate objects in a hazardous environment, which naturally takes disaster hit areas into consideration [10]. The robots can even work together to build a complex 3D structure [10]. Other applications include mining in hazardous areas, agricultural tasks like foraging [10]. A large number of robots can act as an autonomous army. The U.S. Navy has created a swarm of boats which can track an enemy boat, surround it and then destroy it [11]. A fundamental job of a system of

P. Krishnan et al. (Eds.): ICDCIT 2017, LNCS 10109, pp. 15–26, 2017.
DOI: 10.1007/978-3-319-50472-8_2

swarm robots is to move and coordinate their activities for executing some given task, e.g., forming a said pattern or meeting at a location which is the scope of this paper.

1.1 Framework

In the system of swarm robots [2,6] considered in this paper, the robots are distributed in nature. They move and compute of their own independent of other robots. Each robot is capable of sensing or observing (look) its immediate surroundings, performing computations on the sensed data to determine a destination to move to (compute), and moving towards the computed destination (move); its behavior is an endless cycle of sensing, computing, moving and being inactive. However, it can not be inactive for an infinite time. Since the robots are autonomous, there is no centralized mechanism to control them. The robots do not have any common coordinate system. They are asynchronous [6], in the sense that, the amount of time spent in observation, computation, movement and inactivity is finite but not bounded or predictable or same for all the robots. The robots are anonymous, identical in nature and exhibit the same deterministic algorithm which takes the observed positions of the robots within the visibility radius as input and returns a destination point towards which the executing robot moves. After moving to this computed destination, another fresh cycle is initiated and the computed data of the past cycle is removed.

The robots are deployed randomly on the nodes of a graph [1,2]. The movements of the robots are instantaneous, they are always perceived over nodes and not over the links. The robots are memoryless (oblivious) which means they do not have the capacity to remember their previous calculations or observations or leave any mark at the visited nodes. The robots cannot communicate or send messages to each other. Having limited visibility [2,5], the robots cannot observe the entire spatial universe but a part of it which may not contain the positions of all other robots at that time. Thus the robots are not aware of the total number of robots in the system. The robots also can not differentiate between a node having single robot and a node having multiple robots (multiplicity).

The robots are initially randomly placed in the nodes of an anonymous graph. Our objective is to give a deterministic and distributed movement strategy for the robots in order to collect them at a single node without letting the robots to detect multiplicity of nodes. In this paper we have considered an infinite grid graph (where the number of nodes and edges of the graph is not restricted) and a ring graph and proposed two algorithms for gathering of the robots for these two types of graphs.

1.2 Related Works

Gathering in Ring: There have been a large number of researches [2,3,7–9] going on for gathering robots in a ring. D'Angelo et al. [2] have proposed an algorithm to gather asynchronous oblivious robots in a ring with the help of *global* *weak multiplicity*[1] detection capability. D'Angelo et al. [3] have also presented an algorithm to gather six robots on anonymous symmetric rings where *global strong multiplicity*[2] detection capability is used. Kamei et al. [7] have proposed an algorithm to gather asynchronous mobile robots from symmetric configurations without global multiplicity detection capability. However, the authors put forward a gathering protocol for an odd number of robots in a ring-shaped network that allows symmetric but not periodic configurations as initial configurations, using only local weak multiplicity detection capability, where a robot can identify the existence of other robots in its own node. Izumi et al. [5] have proposed an algorithm to gather mobile robots using local weak multiplicity detection capability. Klasing et al. [9] have presented an algorithm to gather asynchronous oblivious robots in a ring, taking advantage of symmetries and global weak multiplicity detection capability. The algorithm provides procedures for gathering all configurations on the ring with more than 18 robots for which gathering is feasible. Various cases of gathering in ring are listed under the section of open problems. So far the only successful results reported are by using multiplicity detection capability. Even multiplicity detection capability does not solve all the cases in ring topology e.g., sp4 problem [9]: presence of 4 robots in a 5-node ring where each node is occupied by one robot initially, rendezvous problem [2]: gathering two robots in a ring.

Our Contribution (I): All the proposed results for successful gathering in a ring are based on the multiplicity detection capabilities of the robots. In this paper we have investigated an approach to get rid of this assumption. Moreover, we also restrict the robots in terms of their visibility range. The robots can not see the complete ring at a glance. Instead a robot can capture $\lceil \frac{N}{2} \rceil$ nodes (where N is the number of nodes of the ring) at its both sides and combines the results to get a complete ring. It is true that if we club two $\lceil \frac{N}{2} \rceil$ strings then it becomes full visibility, however, $\lceil \frac{N}{2} \rceil$ strategy can be implemented using less powerful cameras of robots in practical sense and also it can reduce the number of computations on collected data by the robots. We also consider that the robots agree on orientation of the ring. Without this assumption it is not possible to solve 2-node problem where only two nodes of a ring is occupied by the robots and all other nodes are free. The robots are strictly asynchronous, i.e., there exists always a robot which is not in the same phase of cycle with the others. Taking advantages of $\lceil \frac{N}{2} \rceil$ visibility, orientation of the ring and asynchrony of the robots, our algorithm is able to gather the robots in all configurations and any number of robots >1 and without any extra assumption. It also successfully works for sp4 problem [9] and the rendezvous problem [8]. Section 2 presents an algorithm for gathering in ring and its proof of correctness.

Gathering in Grid: A complete characterization of gathering in finite grid have been reported by D'Angelo et al. [1]. For finite grid there exists some special

[1] The robots can identify the node with multiple robots but can not count the number of robots.

[2] The robots can count the number of robots in a node.

vertices like corner vertices which play an important role to fix the gathering node. However, even with full visibility and multiplicity detection capabilities of the robots, gathering is still not possible in some specific configurations in finite grid. In case of infinite grid computing a gathering node is much more challenging, since there is no boundary. Di Stefano et al. [4] have presented a full characterization for optimal gathering in infinite grid where the robots are able to observe all other robots in the system.

Our Contribution (II): All the reported results on gathering in grid consider that the robots can see all other robots. In this paper we have proposed an algorithm where the robots can see up to a certain radius, 2 node hop distance around itself for this paper. The robots do not know the total number of robots. They also can not detect the multiplicity of the nodes. The infinite grid is embedded in a Cartesian plane. The robots agree on common direction and orientation of the coordinate axes. The algorithm works for any scheduling of the cycle of the robots. Section 3 presents an algorithm for gathering in grid and its proof of correctness.

2 Gathering in Ring

We consider a set of autonomous mobile robots endowed with visibility sensors and motion actuators but that are otherwise unable to communicate. The robots initially randomly deployed in the nodes of a ring graph. The robots exhibit the characteristics of being anonymous, oblivious, silent, asynchronous, and having $\lceil \frac{N}{2} \rceil$ visibility in both sides (N is the number of nodes of the ring). The ring graph is disoriented in nature and the robots reside in the nodes. Neither the nodes nor the edges or the links between the nodes have any label. The robots can be only perceived in the nodes and not over the edges or the links. The robots have to gather at a single node that is unknown beforehand, and to remain there thereafter. Though the ring and the robots are anonymous, for describing the algorithm we have assigned the names n_1, n_2 and so on to the nodes and r_1, r_2 and so on to the robots.

2.1 Algorithm Description

The robots study the occupancy rate on both sides of a robot and then decide on the direction in which the robot should take the next hop. The algorithm is as follows: -

- Step 1: - A robot generates two strings of length $\lceil \frac{N}{2} \rceil$ from its both directions. If the robot finds a node to be occupied then it puts a 1 against that node in the string or if the robot finds a node to be unoccupied then it puts a 0 against that node in the string. For example, in Fig. 1 the strings generated by r_1 are $\{0, 1, 1, 1\}$ in clockwise direction and $\{1, 0, 0, 1\}$ in anticlockwise direction.
- Step 2: - The robot checks the occupancy rate on both sides i.e. the number of nodes occupied by other robots on each side. The robot may do so simply by counting the number of 1s and 0s in the strings.

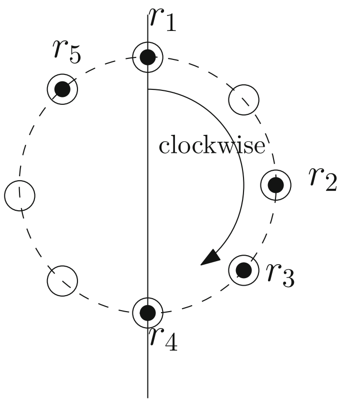

Fig. 1. An example of occupancy rate study

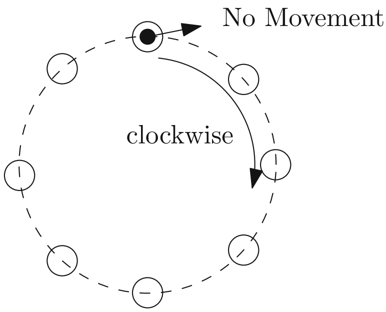

Fig. 2. Case 3.1.

– Step 3: - This is the most important step since in this step the robot makes a decision on its movement. The cases are listed as follows: -
  - Case 3.1: - If the occupancy rate is nil on both sides then the robot does not make any movement (Fig. 2).
  - Case 3.2: - If the occupancy rate is equal on both sides then the robot makes one hop movement to the side with closer neighboring occupied nodes (Fig. 3(a)) or any of the sides if there is a tie (Fig. 3(b)).
  - Case 3.3: - If the occupancy rate is more on the counter-clockwise direction but the clockwise string is nil then the robot does not make any movement (Fig. 4(a)) else it makes one hop movement to the counter-clockwise direction (Fig. 4(b)).
  - Case 3.4: - If the occupancy rate is more on the clockwise direction then the robot makes one hop movement to the clockwise direction (Fig. 5).

2-Node Problem: The main concern which may act as a thorn on the path of gathering is the 2-node problem, where two nodes in the ring are occupied by

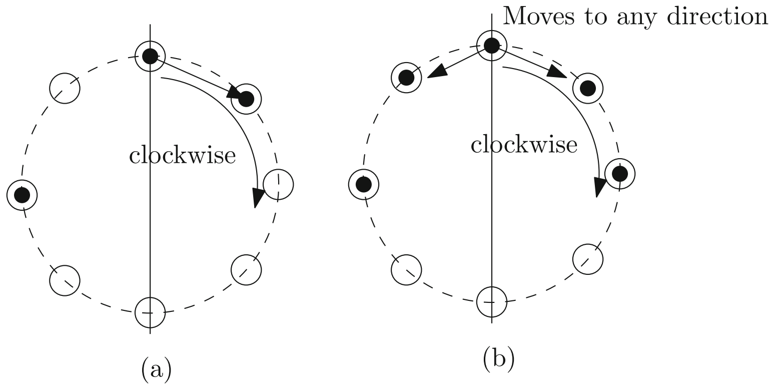

Fig. 3. Case 3.2.

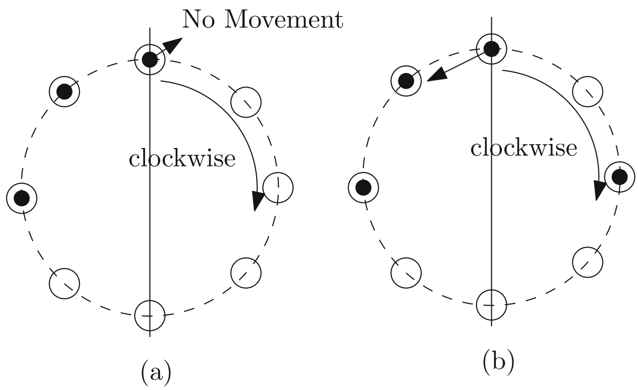

Fig. 4. Case 3.3.

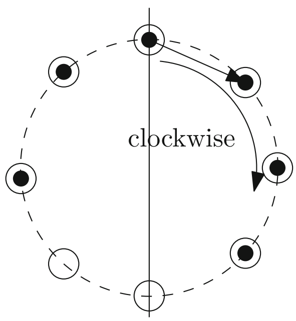

Fig. 5. Case 3.4.

the robots. In fact the inclusion of the property of chirality in the algorithm is just to tackle the 2-node problem otherwise $\lceil \frac{N}{2} \rceil$ visibility range alone would be enough for the gathering to complete. So we put a special condition in case 3.3 under step 3 to counter the 2-node instability. If a robots finds more occupied nodes on the counter-clockwise direction but the clockwise string is nil then the robot does not make any movement else it makes one hop movement to the counter-clockwise direction. This is done to check the increment in the number of nodes between two gathering points or rather in 2-node problem as shown in Fig. 6.

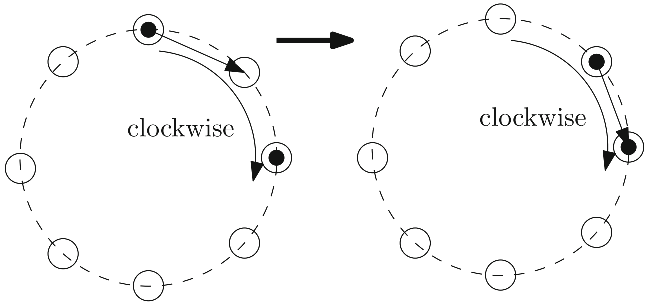

Fig. 6. An example of 2-node problem

2.2 Correctness

The algorithm has been studied extensively in strictly asynchronous model where there exists always a robot which is out of phase of the others. The correctness of the algorithm is proved by the following facts.

- The robot movements are always towards a congested area of the ring and maintain this congestion.
- The robots keep executing the cycles of movement till it finds both half sides of it having only empty nodes.
- The agreement in orientation helps the robots to get into non-repeated configurations by putting restrictions in their movements.

Following lemmas are presented to support these facts.

Lemma 1. *All robots will gather at a single node.*

Proof. According to the algorithm, a robot continues to move until and unless it finds the occupancy rate in both the strings to be nil. This may only happen when all the robots have gathered on a single node.

This lemma also stands true for 2-node problem where the gathering forms at two nodes but the ultimate task is to gather all the robots at a single node.

□

Lemma 2. *The algorithm guarantees progress, i.e., there will be no deadlock.*

Proof. According to the algorithm, a robot does not make a movement only in a situation where each robot finds both its strings to be nil, i.e., gathering is achieved. In all other situation the robots necessarily make one hop movement to any side in all the cycles as per step 3. Hence, there will be no deadlock. □

Lemma 3. *Gathering takes place in finite time.*

Proof. For contradiction, let us assume that the gathering takes place in infinite time. Let $t = \{t_1, t_2, \ldots, t_T\}$ be the set of time intervals. Let $n = \{n_1, n_2, \ldots, n_N\}$ be the set of ring nodes. Let $r = \{r_1, r_2, \ldots, r_R\}$ be the set of robots. So as per our assumption it can be said that gathering will not take place even in t_T time intervals where, $T = 1, 2, \ldots$ But according to the algorithm, the tendency of the robots is to move towards the direction where the occupancy rate or the number of occupied nodes is more. Let the number of empty nodes between the pair of robots at the maximum distance be $N1$. After the first iteration i.e., at the end of t_1 time interval, the number of empty nodes (distance) on the more occupied side between the pair of robots at the maximum distance will be maximum $N1-1$ or less. This distance is decreasing as the robots move. This implies that within t_T time intervals, gathering is achieved where, T is finite. Here we get contradiction. Therefore, the algorithm confirms gathering in finite time. □

3 Gathering in an Infinite Grid

Given an infinite grid G. We assume that the infinite grid is embedded in a Cartesian plane. A set of robots R is deployed in various nodes of G. The robots in R have to gather at a single node in G. This gathering node is not given in advance. The robots agree on directions of coordinate axes. Any robot can observe its 360 degree surrounding up to 2 node hop distance. For example in Fig. 7, r_i can not see any robot outside the dotted circle. A *visibility graph* (G_R) is defined as: *The robot positions are considered as the nodes of* G_R*; If two robots* r_i *and* r_j *are visible to each other, there exists an edge between* r_i *and* r_j *in* G_R.

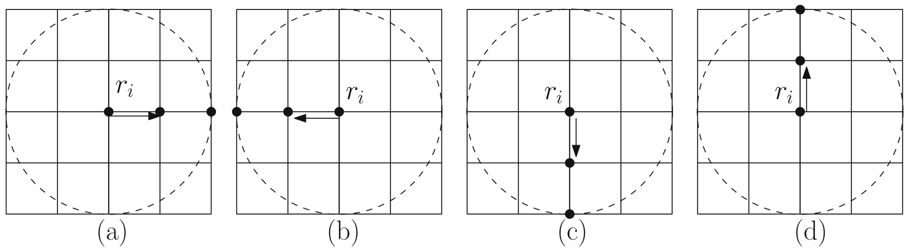

Fig. 7. Case 1.

3.1 Algorithm Description

The proposed algorithm for gathering is based on following two assumptions.

- Initially G_R is connected.
- Initially G_R is acyclic.

The robots keep these assumptions invariant throughout the algorithm. According to the algorithm different movement strategies for the robots are described as below.

- Case 1: If a robot r_i in R finds robots in two consecutive nodes at its east or west or south or north (any one side), it moves one hop to that side (Fig. 7).
- Case 2: If a robot r_i in R finds robot only in its adjacent node at east or south (any one side), it moves to that node (Fig. 8(a), (c)).
- Case 3: If a robot r_i in R finds robot in its east or west or south or north (any one side) which is not adjacent to r_i, it moves one hop to that side (Fig. 9).
- Case 4:
  - a: If a robot r_i in R finds robot only in its south-east or east-north (any one side), it moves one hop to east side (Fig. 10(a), (b)).
  - b: If a robot r_i in R finds robot only in its north-west or west-south (any one side), it moves one hop to west side (Fig. 10(c), (d)).
- Case 5: For all other cases r_i does not move.

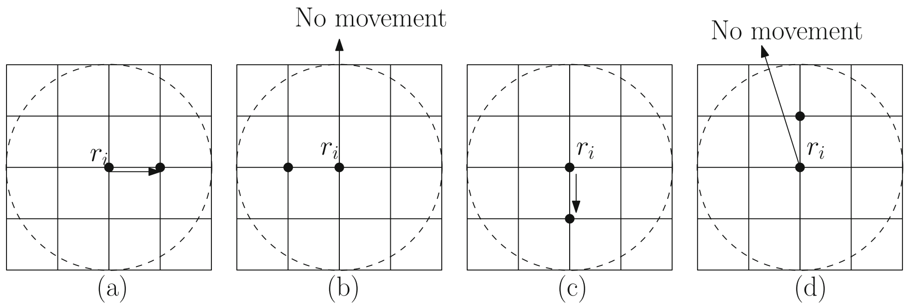

Fig. 8. Case 2.

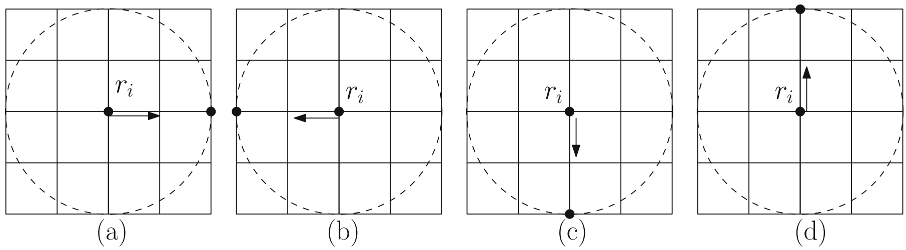

Fig. 9. Case 3.

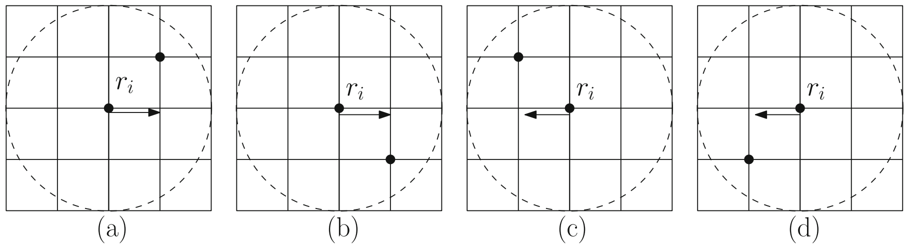

Fig. 10. Case 4.

3.2 Correctness

The correctness of the algorithm is followed from the following facts:

- The assumptions on connectivity and acyclic features of G_R remain as invariant till the robots gather.
- There exists always a robot in G_R eligible for movements till the robots gather.
- The number of nodes in G_R is strictly decreasing by the robot's movements. Finally the number of nodes in G_R becomes *one*, i.e., the robots gather.

Following lemmas are presented to prove these facts.

Lemma 4. *G_R will be connected throughout the algorithm.*

Proof. A robot is selected for movement only if it has degree one. It always moves towards a node occupied with other robots. Thus it does not loose any connectivity. G_R remains connected. □

Lemma 5. *G_R will be acyclic throughout the algorithm.*

Proof. A single degree robot is merging with its immediate neighbor. It does not create any new edge connection. Moreover, the robot with which the moving robot is getting merged is not a part of any cycle. So cycle formation is not possible here. Thus, G_R will be acyclic throughout the algorithm. □

Lemma 6. *The proposed algorithm for gathering in infinite grid guarantees progress.*

Proof. Since G_R is always cycle free, there exists a robot of degree one which moves, unless the robots are already gathered. Hence, progress is assured.

Lemma 7. *The robots gather using finite number of movements.*

Proof. According to the algorithm by case 1, 2, and 4 the robots always move to a node occupied by other robots. Case 3 changes the configuration to case 2. (i) Thus, the movements of the robots reduce the number of robot positions on the grid. Moreover, the visibility graph of the robots remains connected (Lemma 4). (ii) From Lemma 6, there exists always a robot eligible for movement till the robots gather. (i) and (ii) together imply that the number of nodes in G_R strictly decreases by movements of the robots. Thus the number of nodes in G_R becomes *one* in a finite number of movements of the robots, i.e., the robots gather using finite number of movements. □

4 Conclusion

In this paper we have proposed two algorithms, one for gathering robots in a ring and another for gathering robots in an infinite grid. Both the cases open up new scopes for continuation:

- It would be interesting to explore if orientation can be removed to solve gathering in ring in general or clubbed with any other minimal number of additional characteristics. Effort can be made to restrict the vision of the robots as much as possible, e.g., one hop visibility range. Developing optimal gathering algorithm by minimizing the number of movements, is another open direction.
- In the case of grid, the proposed algorithm works under axes agreement. The continuation of this work will be to remove this assumption if possible or to prove that without axes agreement gathering is not possible without any other assumptions. Removing the assumption on cyclic visibility graph may be another immediate future work.

References

1. D'Angelo, G., Di Stefano, G., Klasing, R., Navarra, A.: Gathering of robots on anonymous grids and trees without multiplicity detection. Theoret. Comput. Sci. **610**, 158–168 (2016). Part B
2. D'Angelo, G., Stefano, G., Navarra, A.: How to gather asynchronous oblivious robots on anonymous rings. In: Aguilera, M.K. (ed.) DISC 2012. LNCS, vol. 7611, pp. 326–340. Springer, Heidelberg (2012). doi:10.1007/978-3-642-33651-5_23
3. D'Angelo, G., Stefano, G., Navarra, A.: Gathering of six robots on anonymous symmetric rings. In: Kosowski, A., Yamashita, M. (eds.) SIROCCO 2011. LNCS, vol. 6796, pp. 174–185. Springer, Heidelberg (2011). doi:10.1007/978-3-642-22212-2_16
4. Di Stefano, G., Navarra, A.: Optimal gathering on infinite grids. In: Felber, P., Garg, V. (eds.) SSS 2014. LNCS, vol. 8756, pp. 211–225. Springer, Heidelberg (2014). doi:10.1007/978-3-319-11764-5_15
5. Izumi, T., Izumi, T., Kamei, S., Ooshita, F.: Mobile robots gathering algorithm with local weak multiplicity in rings. In: Patt-Shamir, B., Ekim, T. (eds.) SIROCCO 2010. LNCS, vol. 6058, pp. 101–113. Springer, Heidelberg (2010). doi:10.1007/978-3-642-13284-1_9
6. Kamei, S., Lamani, A., Ooshita, F., Tixeuil, S.: Asynchronous mobile robot gathering from symmetric configurations without global multiplicity detection. In: Kosowski, A., Yamashita, M. (eds.) SIROCCO 2011. LNCS, vol. 6796, pp. 150–161. Springer, Heidelberg (2011). doi:10.1007/978-3-642-22212-2_14. Proceedings of the 37th International Symposium on Mathematical Foundations of Computer Science (MFCS), vol. 7464, pp. 542–553. Springer, Heidelberg (2012)
7. Klasing, R., Kosowski, A., Navarra, A.: Taking advantage of symmetries: gathering of many asynchronous oblivious robots on a ring. Theoret. Comput. Sci. **411**, 3235–3246 (2010)
8. Klasing, R., Markou, E., Pelc, A.: Gathering asynchronous oblivious mobile robots in a ring. Theoret. Comput. Sci. **390**, 27–39 (2008)

9. Koren, M., Klasing, R., Markou, E., Pelc, A.: Gathering asynchronous oblivious mobile robots in a ring. Theoret. Comput. Sci. **390**(1), 27–39 (2008)
10. Michael, N., Fink, J., Kumar, V.: Cooperative manipulation and transportation with aerial robots. Auton. Robots **30**(1), 73–86 (2011)
11. Steinberg, M.: Intelligent autonomy for unmanned naval systems. In: In Defense and Security Symposium, p. 623013. International Society for Optics and Photonics (2006)

http://www.springer.com/978-3-319-50471-1

Distributed Computing and Internet Technology
13th International Conference, ICDCIT 2017,
Bhubaneswar, India, January 13-16, 2017, Proceedings
Krishnan, P.; Radha Krishna, P.; Parida, L. (Eds.)
2017, XIX, 239 p. 87 illus., Softcover
ISBN: 978-3-319-50471-1