Rendezvous of Asynchronous Mobile Robots with Lights

Takashi OKUMURA[1], Koichi WADA[2], and Yoshiaki KATAYAMA[3]

[1] Graduate School of Science and Engineering, Hosei University,
Tokyo 184-8584 Japan.
takashi.okumura.4e@stu.hosei.ac.jp

[2] Faculty of Science and Engineering, Hosei University,
Tokyo, 184-8485, Japan.
wada@hosei.ac.jp

[3] Graduate School of Engineering, Nagoya Institute of Technology,
Nagoya, 466-8555, Japan.
katayama@nitech.ac.jp

Abstract. We study a *Rendezvous* problem for 2 autonomous mobile robots in asynchronous settings with persistent memory called *light*. It is well known that Rendezvous is impossible when robots have no lights in basic common models, even if the system is semi-synchronous. On the other hand, Rendezvous is possible if robots have lights with a constant number of colors in several types of lights[9, 21]. In asynchronous settings, Rendezvous can be solved by robots with 3 colors of lights in non-rigid movement and with 2 colors of lights in rigid movement, respectively[21], if robots can use not only own light but also other robot's light (*full-light*), where non-rigid movement means robots may be stopped before reaching the computed destination but can move a minimum distance $\delta > 0$ and rigid movement means robots can reach the computed destination. In semi-synchronous settings, Rendezvous can be solved with 2 colors of full-lights in non-rigid movement.

In this paper, we show that in asynchronous settings, Rendezvous can be solved with 2 colors of full-lights in non-rigid movement if robots know the value of the minimum distance δ. We also show that Rendezvous can be solved with 2 colors of full-lights in general non-rigid movement if we consider some reasonable restricted class of asynchronous settings.

1 Introduction

Background and Motivation

The computational issues of autonomous mobile robots have been research object in distributed computing fields. In particular, a large amount of work has been dedicated to the research of theoretical models of autonomous mobile robots [1–3, 6, 12, 15, 18, 19]. In the basic common setting, a robot is modeled as a point in a two dimensional plane and its capability is quite weak. We usually assume that robots are *oblivious* (no memory to record past history), *anonymous* and *uniform* (robots have no IDs and run identical algorithms)[8].

Robots operate in Look-Compute-Move (LCM) cycles in the model. In the Look operation, robots obtain a snapshot of the environment and they execute the same algorithm with the snapshot as an input in the Compute operation, and move towards the computed destination in the Move operation. Repeating these cycles, all robots perform a given task. It is difficult for these too weak robot systems to accomplish the task to be completed. Revealing the weakest capability of robots to attain a given task is one of the most interesting challenges in the theoretical research of autonomous mobile robots.

The problem considered in this paper is *Gathering*, which is one of the most fundamental tasks of autonomous mobile robots. Gathering is the process of n mobile robots, initially located on arbitrary positions, meeting within finite time at a location, not known a priori. When there are two robots in this setting, this task is called *Rendezvous*. In this paper, we focus on Rendezvous in asynchronous settings and we reveal the weakest additional assumptions for Rendezvous.

Since Gathering and Rendezvous are simple but essential problems, they have been intensively studied and a number of possibility and/or impossibility results have been shown under the different assumptions[1–3, 5–7, 10, 13–18]. The solvability of Gathering and Rendezvous depends on the activation schedule and the synchronization level. Usually three basic types of schedulers are identified, the fully synchronous (FSYNC), the semi-synchronous (SSYNC) and the asynchronous (ASYNC). Gathering and Rendezvous are trivially solvable in FSYNC and the basic model. However, these problems can not be solved in SSYNC without any additional assumptions [8].

In [4], persistent memory called *light* has been introduced to reveal relationship between ASYNC and SSYNC and they show asynchronous robots with lights equipped with a constant number of colors, are strictly more powerful than semi-synchronous robots without lights. In order to solve Rendezvous without any other additional assumptions, robots with lights have been introduced[4, 9, 21]. Table 1 shows results to solve Rendezvous by robots with lights in each scheduler and movement restriction. In the table, *full-light* means that robots can see not only lights of other robots but also their own light, and *external-light* and *internal-light* mean that they can see only lights of other robots and only own light, respectively. In the movement restriction, Rigid means that robots can reach the computed destination. In Non-Rigid, robots may be stopped before reaching the computed destination but move a minimum distance $\delta > 0$. Non-Rigid($+\delta$) means it is Non-Rigid and robots know the value δ. The Gathering of robots with lights is discussed in [20].

Our Contribution

In this paper, we consider whether we can solve Rendezvous in ASYNC with the optimal number of colors of light. In SSYNC, Rendezvous cannot be solved with one color but can be solved with 2 colors in Non-Rigid and full-light. On the other hand, Rendezvous in ASYNC can be solved with 3 colors in Non-Rigid and full-light, with 3 colors in Non-Rigid($+\delta$) and external-light, or with 12 colors in Rigid and internal-light, respectively.

Table 1. Rendezvous algorithms by robots with lights.

scheduler	movement	full-light[21]	external-light[9]	internal-light[9]	no-light[8, 17]
FSYNC	Non-Rigid	\	\	\	○
SSYNC	Non-Rigid	2	3	?	×
	Rigid	\	?	6	
	Non-Rigid($+\delta$)	\	?	3	
ASYNC	Non-Rigid	3	?	?	×
	Rigid	2	12	?	
	Non-Rigid($+\delta$)	?	3	?	

Back slash indicates that this part has been solved in a weaker condition.
? means this part is not solved.

In this paper we consider Rendezvous algorithms in ASYNC with the optimal number of colors of light and we show that Rendezvous in ASYNC can be solved with 2 colors in Rigid and full-light, or in Non-Rigid($+\delta$) and full-light. We give a basic Rendezvous algorithm with 2 colors of full-lights (A and B) and it can solve Rendezvous in ASYNC and Rigid and its variant can also solve Rendezvous in ASYNC and Non-Rigid($+\delta$). These two algorithms can behave correctly if the initial color of each robot is A. However if the initial color of each robot is B, the algorithm cannot solve Rendezvous in ASYNC and Rigid. It is still open whether Rendezvous can be solved with 2 colors in ASYNC and Non-Rigid, however we introduce some restricted class of ASYNC called *LC-atomic* and we show that our basic algorithm can solve Rendezvous in this scheduler and Non-Rigid with arbitrary initial color, where LC-atomic ASYNC means we consider from the beginning of each Look operation to the end of the corresponding Compute operation as an atomic one, that is, any robot cannot observe between the beginning of each Look operation and the end of each Compute one in every cycle. This is a reasonable sufficient condition Rendezvous is solved with the optimal number of colors of light in ASYNC and No-Rigid.

2 Model and Preliminaries

We consider a set of n anonymous mobile robots $\mathcal{R} = \{r_1, \ldots, r_n\}$ located in $\mathbb{R}^2$. Each robot r_i has a persistent state $\ell(r_i)$ called *light* which may be taken from a finite set of colors L.

We denote by $\ell(r_i, t)$ the color of light the robot r_i has at time t and $p(r_i, t) \in \mathbb{R}^2$ the position occupied by r_i at time t represented in some global coordinate system. Given two points $p, q \in \mathbb{R}^2$, $dis(p, q)$ denotes the distance between p and q.

Each robot r_i has its own coordinate system where r_i is located at its origin at any time. These coordinate systems do not necessarily agree with those of other robots. It means that there is no common unit of distance and no common knowledge of directions of its coordinates and clockwise orientation (*chirality*).

At any point of time, a robot can be active or inactive. When a robot r_i is activated, it executes Look-Compute-Move operations:

- **Look:** The robot r_i activates its sensors to obtain a snapshot which consists of pairs of a light and a position for every robot with respect to its own coordinate system. We assume robots can observe all other robots(unlimited visibility).
- **Compute:** The robot r_i executes its algorithm using the snapshot and its own color of light (if it can be utilized) and returns a destination point des_i by its coordinate system and a light $\ell_i \in L$ to which its own color is set.
- **Move:** The robot r_i moves to the computed destination des_i. The robot may be stopped by an adversary before reaching the computed destination. If stopped before reaching its destination, a robot moves at least a minimum distance $\delta > 0$. If the distance to the destination is at most δ, the robot can reach it. In this case, the movement is called *Non-Rigid*. Otherwise, it is called *Rigid*. If the movement is Non-Rigid and robots know the value of δ, it is called *Non-Rigid(+δ)*.

A scheduler decides which subset of robots is activated for every configuration. The schedulers we consider are asynchronous and semi-synchronous and it is assumed that schedulers are *fair*, each robot is activated infinitely often.

- **ASYNC:** The asynchronous (ASYNC) scheduler, activates the robots independently, and the duration of each Compute, Move and between successive activities is finite and unpredictable. As a result, robots can be seen while moving and the snapshot and its actual configuration are not the same and so its computation may be done with the old configuration.
- **SSYNC:** The semi-synchronous(SSYNC) scheduler activates a subset of all robots synchronously and their Look-Compute-Move cycles are performed at the same time. We can assume that activated robots at the same time obtain the same snapshot and their Compute and Move are executed instantaneously. In SSYNC, we can assume that each activation defines discrete time called *round* and Look-Compute-Move is performed instantaneously in one round.

As a special case of SSYNC, if all robots are activated in each round, the scheduler is called full-synchronous (FSYNC).

In this paper, we consider ASYNC and we assume the followings;

In a Look operation, a snapshot of a time t_L is taken and we say that *Look operation is performed at time t_L*. Each Compute operation of r_i is assumed to be done at an instant time t_C and its color of light $\ell_i(t)$ and its destination des_i are assigned to the computed values at the time t_C. In a Move operation, when its movement begins at t_B and ends at t_E, we say that its movement is performed during $[t_B.t_E]$, its beginning and ending of the movement are denoted by $Move_{BEGIN}$ and $Move_{END}$, and its $Move_{BEGIN}$ and $Move_{END}$ occur at t_B and t_E, respectively. In the following, $Compute$, $Move_{BEGIN}$ and $Move_{END}$ are abbreviated as $Comp$, $Move_B$ and $Move_E$, respectively. When some cycle has no movement (robots change only colors of lights, or their destinations are the current positions), we can assume the Move operation in this cycle is omitted, since we can consider the Move operation can be performed just before the next Look operation.

Also we consider the following restricted classes of ASYNC;

Let a robot execute a cycle. If any other robot cannot execute any Look operation between the Look operation and the following Compute one in the cycle, its ASYNC is said to be *LC-atomic*. Thus we can assume that in LC-atomic ASYNC, Look and Compute operations in every cycle are performed at the same time. If any other robot cannot execute any Look operation between the $Move_B$ and the following $Move_E$, its ASYNC is said to be *$Move$-atomic*. In this case Move operations in all cycles can be considered to be performed instantaneously and at time t_M. In Move-atomic ASYNC, when a robot r observes another robot r' performing a Move operation at time t_M, r observes the snapshot after the moving of r'.

In our settings, robots have persistent lights and can change their colors at an instant time in each Compute operation. We consider the following robot models according to visibility of lights.

- *full-light*, the robot can recognize not only colors of lights of other robots but also its own color of light.
- *external-light*, the robot can recognize only colors of lights of other robots but cannot see its own color of light. Note robot can change its own color.
- *internal-light*, the robot can recognize only its own color of light but cannot see colors of lights of other robots.

An *n-Gathering* problem is defined that given $n(\geq 2)$ robots initially placed in arbitrary positions in $\mathbb{R}^2$, they congregate at a single location which is not predefined in finite time. In the following, we consider the case that $n = 2$ and the 2-Gathering problem is called *Rendezvous*.

3 Previous Results for Rendezvous

Theorem 1. [8] *Rendezvous is deterministically unsolvable in SSYNC even if chirality is assumed.*

If robots have a constant number of colors in their lights, Rendezvous can be solved shown in Table 1.

It is still an open problem that Rendezvous is solved in ASYNC with 2 colors. In the following, we will show that Rendezvous is solved in ASYNC and full-light with 2 colors, if we assume (1) Rigid movement, (2) Non-Rigid movement and knowledge of the minimum distance δ robots move, (3) LC-atomic. In these cases, we can construct optimal Rendezvous algorithms with respect to the number of colors in ASYNC.

4 Asynchronous Rendezvous Algorithms for Robots with Lights

4.1 Basic Rendezvous Algorithm

In this section, two robots are denoted as r and s. Let t_0 be the starting time of the algorithm.

Given a robot *robot*, an operation *op*$(\in \{Look, Comp, Move_B, Move_E\})$, and a time t, $t^+(robot, op)$ denotes the time *robot* performs *op* immediately after t if there exists such operation, and $t^-(robot, op)$ denotes the time *robot* performs *op* immediately before t if there exists such operation. If t is the time the algorithm terminates, $t^+(robot, op)$ is not defined for any *op*. When *robot* does not perform *op* before t and $t^-(robot, op)$ does not exist, $t^-(robot, op)$ is defined to be t_0.

A time t_c is called a *cycle start time*, if the next performed operations of both r and s after t are Look ones, or otherwise, the robots performing the operations neither change their colors of lights nor move. In the latter case, we can consider that these operations can be performed before t_c and the subsequent *Look* operation can be performed as the first operation after t_c.

Algorithm 1 Rendezvous (scheduler, movement, initial-light)

```
Parameters: scheduler, movement-restriction, Initial-light
Assumptions: full-light, two colors ($A$ and $B$)
case me.light of
$A$:
    if other.light $=A$ then
        $me.light \leftarrow B$
        $me.des \leftarrow$ the midpoint of $me.position$ and $other.position$
    else $me.des \leftarrow other.position$
$B$:
    if $other.light = A$ then
        $me.des \leftarrow me.position$ // stay
    else $me.light \leftarrow A$
endcase
```

Algorithm 1 is used as a basic Rendezvous algorithm which has three parameters, schedulers, movement restriction and an initial color of light and assumes full-light and uses two colors A and B.

We will show that Rendezvous(ASYNC, Rigid, A) and Rendezvous(LC-atomic ASYNC, Non-Rigid, any) solve Rendezvous and some variant of Rendezvous(ASYNC, Non-Rigid($+\delta$), A) also solves Rendezvous.

Lemma 1. *Assume that time t_c is a cycle start time and $\ell(r, t_c) = \ell(s, t_c) = B$ in Rendezvous(ASYNC, Non-Rigid, any). If $dis(p(r, t_c)), p(s, t_c)) = 0$, then two robots r and s do not move after t_c.*

Lemma 2. *Let robot r perform Look operation at time t in Rendezvous(ASYNC, Non-Rigid, any). If $t^{-}(s, Comp) \leq t$ and $\ell(r,t) \neq \ell(s,t)$, then there exists a time $t^{*}(> t)$ such that r and s succeed in rendezvous at time t^{*} by Rendezvous(ASYNC, Non-Rigid, any).*

ASYNC and Rigid movement

If Rigid movement is assumed, asynchronous Rendezvous can be done with 2 colors.

Theorem 2. *Rendezvous(ASYNC, Rigid, A) solves Rendezvous.*

Lemma 3. *Rendezvous(LC-atomic ASYNC, Rigid, B) solves Rendezvous.*

Since it is easily verified that there is a cycle start time $t_c(\geq t_0)$ such that $\ell(r,t_c) = \ell(s,t_c) = B$ in an execution of Rendezvous(Async, Non-Rigid, A), it cannot solve Rendezvous even if both initial colors of lights are A. In the next subsection, we will show if ASYNC is restricted to LC-atomic one, Rendezvous can be solved in Non-Rigid with two colors from any initial colors of lights.

LC-atomic ASYNC and Non-Rigid movement

Let t_c be a cycle start time of the algorithm. There are three cases according to the colors of lights of two robots r and s, (I) $\ell(r,t_c) \neq \ell(s,t_c)$, (II) $\ell(r,t_c) = \ell(s,t_c) = A$, and (III) $\ell(r,t_c) = \ell(s,t_c) = B$

Lemma 4. *If $\ell(r,t_c) \neq \ell(s,t_c)$ and the algorithm starts at t_c, then there exists a time $t^{*}(\geq t)$ such that r and s succeed in rendezvous at time t^{*} by Rendezvous(LC-atomic ASYNC, Non-Rigid, any).*

Lemma 5. *If $\ell(r,t_c) = \ell(s,t_c) = B$ and the algorithm starts at t_c, then there exists a time $t^{*}(\geq t)$ such that r and s succeed in rendezvous at time t^{*} by Rendezvous(LC-atomic ASYNC, Non-Rigid, any) or t^{*} is a cycle start time and $\ell(r,t^{*}) = \ell(s,t^{*}) = A$.*

Lemma 6. *If $\ell(r,t_c) = \ell(s,t_c) = A$ and the algorithm starts at t_c, then there exists a time $t^{*}(\geq t_c)$ such that r and s succeed in rendezvous at time t^{*} by Rendezvous(LC-atomic ASYNC, Non-Rigid, any) or t^{*} is a cycle start time, $\ell(r,t^{*}) = \ell(s,t^{*}) = A$ and $dis(p(r,t^{*}), p(s,t^{*})) \leq dis(p(r,t_c), p(s,t_c)) - 2\delta$.*

Lemmas 3-5 is followed by the next theorem.

Theorem 3. *Rendezvous(LC-atomic ASYNC, Non-Rigid, any) solves Rendezvous.*

ASYNC and Non-Rigid movement($+\delta$)

Although it is still open whether asynchronous Rendezvous can not be solved in Non-rigid with two colors of lights, if we assume Non-Rigid($+\delta$), we can solve Rendezvous modifying Rendezvous(ASYNC, Non-Rigid($+\delta$), A) and using the minimum moving value δ in it.

Algorithm 2 RendezvousWithDelta (ASYNC, Non-Rigid($+\delta$), A)

Assumptions: full-light, two colors (A and B)

```
1:  case $dis(me.position, other.position)(= DIST)$ of
2:  $DIST > 2\delta$:
3:      if me.light =other.light =$B$ then
4:          $me.des \leftarrow$ the point moving by $\delta/2$ from $me.position$ to $other.position$
5:      else $me.light \leftarrow B$
6:  $2\delta \geq DIST \geq \delta$:
7:      if $me.light = other.light = A$ then
8:          $me.light \leftarrow B$
9:          $me.des \leftarrow$ the midpoint of $me.position$ and $other.position$
10:     else $me.light \leftarrow A$
11: $\delta > DIST$: //Rendezvous(ASYNC, Rigid, A)
12:     case me.light of
13:     $A$:
14:         if other.light =$A$ then
15:             $me.light \leftarrow B$
16:             $me.des \leftarrow$ the midpoint of $me.position$ and $other.position$
17:         else $me.des \leftarrow other.position$
18:     $B$:
19:         if $other.light = A$ then $me.des \leftarrow me.position$ // stay
20:         else $me.light \leftarrow A$
21:     endcase
22: endcase
```

Let $dist_0 = dis(p(r, t_0), p(s, t_0))$ and let RendezvousWithDelta (Algorithm 2) begin with $\ell(r, t_0) = \ell(s, t_0) = A$. If $dist_0 > 2\delta$, both robots do not move until both colors of lights become B(**lines** 3-5) and there exists a cycle start time $t_1(> t_0)$ such that $\ell(r, t_1) = \ell(s, t_1) = B$. After $\ell(r, t_1) = \ell(s, t_1) = B$, the distance between r and s is reduced by $\delta/2$ without changing the colors of lights(**line** 4) and the distance falls in $[2\delta, \delta]$ and both colors of lights become A at a cycle starting time t_2. After $\ell(r, t_2) = \ell(s, t_2) = A$, we can use Rendezvous(ASYNC, Rigid, A) since $2\delta \geq dis(p(r, t_2), p(s, t_2)) \geq \delta$. Therefore, rendezvous is succeeded. Note that in Algorithm 2, the initial pair of colors of r and s is $(\ell(r, t_0), \ell(s, t_0)) = (A, A)$ and it is changed into $(\ell(r, t_1), \ell(s, t_1)) = (B, B)$ without changing the distance of r and s. And it is changed into $(\ell(r, t_2), \ell(s, t_2)) = (A, A)$ when the distance becomes between δ and 2δ. These *mode* changes are necessary and our algorithm does not work correctly, if these mode changes are not incorporated in the algorithm.

Lemma 7. *If $dist_0 > 2\delta$, in any execution of RedezvousWithDelta(ASYNC, Non-Rigid($+\delta$), A),*

(1) there exists a cycle start time $t_1(> t_0)$ such that $\ell(r, t_1) = \ell(s, t_1) = B$ and $dis(p(r, t_1), p(s, t_1)) = dist_0$, and

(2) there exists a cycle start time $t_2 (> t_1)$ such that $\ell(r, t_2) = \ell(s, t_2) = A$ and $2\delta \geq dis(p(r, t_2), p(s, t_2)) \geq \delta$.

Lemma 8. *If $2\delta \geq dist_0 \geq \delta$, then RedezvousWithDelta(ASYNC, Non-Rigid($+\delta$), A) solves Rendezvous.*

Lemma 9. *If $dist_0 > \delta$, then RedezvousWithDelta(ASYNC, Non-Rigid($+\delta$), A) solves Rendezvous.*

Theorem 4. *RedezvousWithDelta(ASYNC, Non-Rigid($+\delta$), A) solves Rendezvous.*

5 Concluding Remarks

We have shown that Rendezvous can be solved in ASYNC with the optimal number of colors of lights if Non-Rigid($+\delta$) movement is assumed. We have also shown that Rendezvous can be solved by an $\mathcal{L}$-algorithm in ASYNC and Non-Rigid with the optimal number of colors of lights if ASYNC is LC-atomic. Interesting open problems are whether can Rendezvous be solved in ASYNC and Non-Rigid with 2 colors or not[4], and what condition of ASYNC can $\mathcal{L}$-algorithms be solved in Non-Rigid with 2 colors?

Acknowledgment This work is supported in part by KAKENHI no. 17K00019 and 15K00011.

References

1. N. Agmon and D. Peleg, Fault-tolerant gathering algorithms for autonomous mobile robots, SIAM *Journal on Computing*, 36, 56–82, 2006.
2. Z. Bouzid, S. Das, and S. Tixeuil, Gathering of mobile robots tolerating multiple crash Faults. In Proceedings of 33rd ICDCS, 2013.
3. M. Cieliebak, P. Flocchini, G. Prencipe, and N. Santoro, Distributed computing by mobile robots: Gathering, SIAM *Journal on Computing*, 41(4): 829–879, 2012.
4. S. Das, P. Flocchini , G. Prencipe, N. Santoro , M. Yamashita , Autonomous mobile robots with lights, Theoretical Computer Science, 609, 171–184, 2016.
5. X. Défago, M. Gradinariu Potop-Butucaru, J. Clément, S. Messika, P. Raipin Parvédy: Fault and Byzantine tolerant self-stabilizing mobile robots gathering - Feasibility study -. CoRRabs/1602.05546, 2016.
6. B. Degener, B. Kempkes, T. Langner, F. Meyer auf der Heide, P. Pietrzyk, and R. Wattenhofer, A tight runtime bound for synchronous gathering of autonomous robots with limited visibility, In Proceedings of 23rd ACM SPAA, 139–148, 2011.
7. Y. Dieudonné and F. Petite, Self-stabilizing gathering with strong multiplicity detection, Theoretical Computer Science, 428(13), 2012.
8. P Flocchini, G Prencipe, N Santoro, Distributed Computing by Oblivious Mobile Robots, Synthesis Lectures on Distributed Computing Theory, Morgan & Claypool, 2012.

[4] Very recently it has been solved affirmatively[11].

9. P.Flocchini , N.Santoro , G.Viglietta , M.Yamashita, Rendezvous with Constant Memory, Theoretical Computer Science, 621, 57–72, 2016.
10. P. Flocchini, G. Prencipe, N. Santoro, and P. Widmayer, Gathering of asynchronous robots with limited visibility, Theoretical Computer Science, 337(1-3):147–168, 2005.
11. A. Heriban, X. Défago, S. Tixeuil, Optimally gathering two robots, Research Report HAL Id: hal-01575451, UPMC Sorbonne Universités, Aug. 2017.
12. T. Izumi, Z. Bouzid, S. Tixeuil, and K. Wada, Brief Announcement: The BG-simulation for Byzantine mobile robots, 25th DISC, 330–331, 2011.
13. T Izumi, Y Katayama, N Inuzuka, and K Wada, Gathering Autonomous Mobile Robots with Dynamic Compasses: An Optimal Result, 21stDISC, 298-312, 2007.
14. T. Izumi, S. Souissi, Y. Katayama, N. Inuzuka, X. Défago, K. Wada, and M. Yamashita, The gathering problem for two oblivious robots with unreliable compasses, SIAM *Journal on Computing*, 41(1):26–46, 2012.
15. S. Kamei, A. Lamani, F. Ooshita, and S. Tixeuil, Asynchronous mobile robot gathering from symmetric configurations without global multiplicity detection, 18th SIRICCO, 150–161, 2011.
16. J. Lin, A.S. Morse, and B.D.O. Anderson, The multi-agent rendezvous problem. parts 1 and 2, SIAM *Journal on Control and Optimization,* 46(6):2096–2147, 2007.
17. G. Prencipe, Impossibility of gathering by a set of autonomous mobile robots, Theoretical Computer Science, 384(2-3):222–231, 2007.
18. S. Souissi, X. Défago, and M. Yamashita, Using eventually consistent compasses to gather memory-less mobile robots with limited visibility, ACM Transactions on Autonomous and Adaptive Systems, 4(1):1–27, 2009.
19. I. Suzuki and M. Yamashita, Distributed anonymous mobile robots: Formation of geometric patterns, SIAM *Journal on Computing*, 28, 1347–1363, 1999.
20. S. Terai, K. Wada, and Y. Katayama, Gathering problems for autonomous mobile robots with lights, Technical Report of Wada Labo., Hosei Unoversity, TRW-16-1, 2016.
21. G. Viglietta, Rendezvous of two robots with visible bits, Technical Report arXiv:1211.6039, 2012.