Optimal Rendezvous $\mathcal{L}$-Algorithms for Asynchronous Mobile Robots with External-Lights

Takashi Okumura
Graduate School of Science and Engineering, Hosei University, Tokyo, 184-8485, Japan
takashi.okumura.4e@stu.hosei.ac.jp

Koichi Wada[1]
Faculty of Science and Engineering, Hosei University, Tokyo, 184-8485, Japan
http://ai.k.hosei.ac.jp/staff/wada
wada@hosei.ac.jp

Xavier Défago
School of Computing, Tokyo Institute of Technology, Tokyo, Japan
defago@c.titech.ac.jp
https://orcid.org/0000-0002-2377-205X

Abstract

We study the *Rendezvous* problem for two autonomous mobile robots in asynchronous settings with persistent memory called *light*. It is well known that Rendezvous is impossible in a basic model when robots have no lights, even if the system is semi-synchronous. On the other hand, Rendezvous is possible if robots have lights of various types with a constant number of colors. If robots can observe not only their own lights but also other robots' lights, their lights are called *full-light*. If robots can only observe the state of other robots' lights, the lights are called *external-light*. This paper focuses on robots with external-lights in asynchronous settings and a particular class of algorithms called $\mathcal{L}$-algorithms, where an $\mathcal{L}$-algorithm computes a destination based only on the current colors of observable lights. When considering $\mathcal{L}$-algorithms, Rendezvous can be solved by robots with full-lights and three colors in general asynchronous settings (called ASYNC) and the number of colors is optimal under these assumptions. In contrast, there exist no $\mathcal{L}$-algorithms in ASYNC with external-lights regardless of the number of colors.

In this paper, extending the impossibility result, we show that there exist no $\mathcal{L}$-algorithms in so-called LC-1-Bounded ASYNC with external-lights regardless of the number of colors, where LC-1-Bounded ASYNC is a proper subset of ASYNC and other robots can execute at most one *Look* operation between the *Look* operation of a robot and its subsequent *Compute* operation. We also show that LC-1-Bounded ASYNC is the minimal subclass in which no $\mathcal{L}$-algorithms with external-lights exist. That is, Rendezvous can be solved by $\mathcal{L}$-algorithms using external-lights with a finite number of colors in LC-0-Bounded ASYNC (equivalently LC-atomic ASYNC). Furthermore, we show that the algorithms are optimal in the number of colors they use.

2012 ACM Subject Classification Theory of computation → Distributed algorithms, Computing methodologies → Self-organization

Keywords and phrases Autonomous mobile robots, Rendezvous, Lights, L-algorithms

Digital Object Identifier 10.4230/LIPIcs.OPODIS.2018.24

Acknowledgements This research was partly supported by JSPS KAKENHI No. 17K00019 and by Japan Science and Technology Agency (JST) SICORP.

[1] contact author

22nd International Conference on Principles of Distributed Systems (OPODIS 2018).
Editors: Jiannong Cao, Faith Ellen, Luis Rodrigues, and Bernardo Ferreira; Article No. 24; pp. 24:1–24:16
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

1 Introduction

Background and Motivation. The computational issues of autonomous mobile robots have been the object of much research in the field of distributed computing. In particular, a large amount of work has been dedicated to the research of theoretical models of autonomous mobile robots [1, 2, 3, 6, 12, 15, 19, 20]. 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 using the snapshot as input for the *Compute* operation, and move towards the computed destination in the *Move* operation. Repeating these cycles, all robots collectively perform a given task. The weak capabilities of the robots make it challenging for them to accomplish even simple tasks. Therefore, identifying the minimum (weakest) capabilities that the robots need to complete a given task in a given model constitutes a very interesting and important challenge for the theoretical research on autonomous mobile robots.

This paper considers the problem of *Gathering*, which is one of the most fundamental tasks for autonomous mobile robots. Gathering is the process where n mobile robots, initially located at arbitrary positions, meet within finite time at a location, not known a priori. When there are two robots in this setting (i.e., for $n = 2$), this task is called *Rendezvous*. In this paper, we focus on Rendezvous in asynchronous settings and we reveal the relationship among several assumptions.

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, 2, 3, 5, 6, 7, 9, 13, 14, 15, 16, 18, 19]. The solvability of Gathering and Rendezvous depends on the activation schedule and the synchronization level. Usually three basic types of schedulers are identified, namely, the fully synchronous (FSYNC), the semi-synchronous (SSYNC) and the asynchronous (ASYNC) models. In the FSYNC model, there is a common round and in each round all robots are activated simultaneously and *Compute* and *Move* are done instantaneously. The SSYNC model is the same as FSYNC except that at each round only a subset of the robots are activated, with a fairness guarantee that every robot is activated infinitely often in any infinite execution. In the ASYNC scheduler, there are no restrictions about the notion of time. In particular, *Compute* and *Move* and the interval between them can take any (finite) duration, a robot can be seen while moving, and in the interval between an observation and a corresponding move other robots may have possibly moved several times. Gathering and Rendezvous are trivially solvable in FSYNC in the basic model (e.g., without lights) by using an algorithm that moves to the center of gravity. However, these problems can not be solved in SSYNC without any additional assumptions [8].

Das *et al.* [4] extend the classical model with persistent memory, called *lights*, to reveal the relationship between ASYNC and SSYNC and they show that asynchronous robots equipped with lights and 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 [10, 4, 21]. Table 1 shows previous results including ours to solve Rendezvous by robots with lights, for each scheduler and movement restriction. In the table, *LC*-atomic ASYNC is a subclass of ASYNC, in which we consider from the beginning of each *Look* operation to the end of the corresponding *Compute* operation as an

Table 1 Rendezvous algorithms by robots with lights.

scheduler	movement	full-light	external-light	internal-light	no-light
FSYNC	Non-Rigid	–	–	–	◯ [8]
SSYNC	Non-Rigid	**2**∗(S) [21]	**3**∗(S) [21]	∞∗ [10]	× [8]
	Rigid	–	–	6 [10]	
	Non-Rigid(+δ)	–	–	3 [10]	
LC-atomic ASYNC	Non-Rigid	**2***(S) [17]	?→ **4***(QS),**5***(S)	?	–
	Rigid	–	? → **3***	?	
	Non-Rigid(+δ)	–	?	?	
ASYNC	Non-Rigid	**2**(S) [11],3*(S) [21]	∞∗ [10]	?	–
	Rigid	**2*** [21]	12 [10]	?	
	Non-Rigid(+δ)	–	3 [10]	?	

◯: solvable, ×: unsolvable. ∗: $\mathcal{L}$-algorithm, (S): self-stabilizing,
(QS): quasi-self-stabilizing. – indicates that this part has been solved under weaker conditions or unsolved under stronger ones. A number represents the number of colors used in these algorithms and it is in **boldface** when optimal. ? means that this part has not been solved.

atomic one, that is, no robot can observe between the beginning of each *Look* operation and the end of the next *Compute* on the same robot [17]. Regarding the various kinds of lights, *full-light* means that robots can see their own light as well as that of the other robots, whereas *external-light* and *internal-light* respectively mean that they can see only the lights of the other robots, or only their own light. Regarding the movement restriction, *Rigid* means that the robots can always reach the computed destination during the move operation. *Non-Rigid* means that robots may be stopped before reaching the computed destination but move a minimum distance $\delta > 0$. Non-Rigid(+δ) means it is Non-Rigid and robots know the value δ.

In Table 1, we can see that complete solutions have been obtained for the case of full-lights. However, the cases of external-lights and internal-lights are still insufficiently explored and should be solved.

Our Contribution. In this paper, we are concerned with Rendezvous for robots equipped with external-lights and a particular class of algorithms called $\mathcal{L}$-algorithms. Briefly, an $\mathcal{L}$-algorithm means that each robot (1) always computes a destination on the line connecting the two robots, and (2) using only the observed colors of the lights of the robots.

Algorithms of this class are of interest because they operate also when the coordinate system of a robot is not self-consistent (i.e., it can unpredictably rotate, change its scale or undergo a reflection) [10]. Rendezvous can be solved by an $\mathcal{L}$-algorithm with 3 colors of external-lights in SSYNC [21], but cannot be solved by any $\mathcal{L}$-algorithm with any number of colors of external-lights in ASYNC [10].

In this paper, we reveal the relationship among the number of colors, movement restrictions and initial configurations on $\mathcal{L}$-algorithms with external-lights in asynchronous settings. We introduce subclasses of ASYNC called *LC*-k-Bounded ASYNC ($k \geq 0$), where *LC*-k-Bounded ASYNC is a subclass of ASYNC in which any other robot can execute at most k *Look* operations between the *Look* operation of a robot and its subsequent *Compute* one. When $k = 0$, it is equivalent to *LC*-atomic ASYNC and any *Look* operation and its subsequent *Compute* one can be executed atomically and this interval cannot be observed by any other robots. We show that Rendezvous cannot be solved by any $\mathcal{L}$-algorithm with any number of colors of external-lights in LC-1-Bounded ASYNC and Rendezvous can be solved by $\mathcal{L}$-algorithm with a finite number of colors of external-lights in LC-0-Bounded (LC-atomic) ASYNC. In fact, we give three $\mathcal{L}$-algorithms with external-lights in LC-atomic ASYNC, such that (1) if we may start from a particular initial configuration with the same color, Rendezvous is solved with 3 colors in Rigid, (2) if we start from any initial configuration with the same color (called *quasi-self-stabilizing*), Rendezvous is solved with 4 colors in Non-Rigid, and (3) if we start from any initial configuration (called *self-stabilizing*), Rendezvous is solved with 5 colors and in Non-Rigid. We also show that the numbers of colors used in the three algorithms are optimal in the sense that no $\mathcal{L}$-algorithm with fewer colors can solve Rendezvous. In order to derive the lower bounds we give several essential properties of $\mathcal{L}$-algorithms.

The remainder of the paper is organized as follows. In Section 2, we define the robot model with lights, the Rendezvous problem, and basic terminology. Section 3 reviews previous results on Rendezvous with lights and the impossibility result of $\mathcal{L}$-algorithms with external-lights is extended. Section 4 shows several properties of $\mathcal{L}$-algorithms for Rendezvous with 3 colors of external-lights and Section 5 shows optimal Rendezvous $\mathcal{L}$-algorithms on Asynchronous robots with external-lights. Section 6 concludes the paper.

2 Preliminaries

2.1 Robot Model

We consider a set of n anonymous mobile robots $\mathcal{R} = \{r_1, \dots, r_n\}$ 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 that the light of 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 and $q \in \mathbb{R}^2$, $dis(p, q)$ 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 knowledge of unit of distance, directions of its coordinates, or 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 a pair of light and position for every robot with respect to the coordinate system of r_i. Since the result of this operation is a snapshot of the positions of all robots, the robot does not notice the movement, even if it sees other moving robots. We assume that robots can observe all other robots (unlimited visibility).
- *Compute*: The robot r_i executes its algorithm using the snapshot and the color of its own light (if allowed by the model) and returns a destination point des_i expressed in 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. A robot r is said to *collide* with robot s at time t if $p(r, t) = p(s, t)$ and at time t r is performing *Move*. The collision is *accidental* if r's destination is not $p(r, t)$. Since robots are seen as points, we assume that accidental collisions are immaterial. A moving robot, upon causing an accidental collision, proceeds in its movement without changes, in a “hit-and-run” fashion [8]. 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 $\delta > 0$.

Note that without this assumption an adversary could make it impossible for any robot to ever reach its destination. 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 are concerned with ASYNC and we assume the followings; In a *Look* operation, a snapshot of the environment at time t_L is taken and we say that the *Look operation is performed at time* t_L. Each *Compute* operation of r_i is assumed to be done at time t_C and the color of its light $\ell_i(t)$ and its pending destination des_i are both set to the computed values for any time greater than t_C[2]. In a *Move* operation, when the movement begins at time t_B and ends at t_E, we say that it is performed during interval $[t_B, t_E]$, and the beginning (resp. ending) of the movement is denoted by $Move_{BEGIN}$ (resp. $Move_{END}$) occurring at time t_B (resp. t_E). In the following, *Compute*, $Move_{BEGIN}$ and $Move_{END}$ are abbreviated as *Comp*, M_B and M_E, respectively. When a cycle has no actual movement (i.e., robots only change color and their destinations are the current positions), we can equivalently assume that the *Move* operation in this cycle is omitted, since we can consider the *Move* operation to be performed just before the next *Look* operation.

Without loss of generality, we assume the set of time instants at which the robots start executions of *Look*, *Comp*, M_B and M_E is $\mathbb{N}$.

We also consider the following restricted classes of ASYNC. Let k be a non-negative integer. Let a robot r execute a cycle. If any other robot can execute at most k *Look* operations between the *Look* operation of r and its subsequent *Compute* in that cycle, the model is said to be *LC-k-Bounded*. If $k = 0$, it is said to be *LC-atomic*. Thus we can assume that in the *LC*-atomic ASYNC model, *Look* and *Comp* operations in every cycle are performed simultaneously (or atomically), say at time t_{LC}, and we say that the *LC*-operation is performed at time t_{LC}.

Similarly, if no other robot can execute at most *Look* operations between the operation M_B of r and its corresponding M_E, the model is said to be *Move-k-Bounded*. If $k = 0$, it is said to be *Move-atomic*. In this case *Move* operations in all cycles can be considered to

[2] Note that if some robot performs a *Look* operation at time t_C, then it observes the former color and if it does at time $t_C + \epsilon (\forall \epsilon > 0)$, then it observes the newly computed color.

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'.

Since each operation occurs at integer times, when LC-operation is performed at time t in LC-atomic ASYNC, we can assume that the snapshot at t is obtained at t and the computation completes at $t+1$. Also when *Move*-operation begins (M_B occurs) at time t in *Move*-atomic ASYNC, M_E can be assumed to occur at time $t+1$. Thus, if a robot r observes another robot r' performing a *Move* operation at time t_M, then r observes the snapshot before the moving of r' until and at time t, and the snapshot after the moving of r' from $t_M + 1$.

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

- *full-light*, a robot can observe the lights of other robots as well as its own, and it can also change the color of its own light.
- *external-light*, a robot can observe the light of other robots but not its own. It can however change the color of its own light in a "write-only" manner.
- *internal-light*, a robot can observe and change the color of its own light, but cannot observe the lights of other robots.

2.2 Rendezvous and $\mathcal{L}$-Algorithms

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

When we consider algorithms on robots with lights, we exclude algorithms that solve Rendezvous only starting from initial settings in which robots have different colors of lights. That is, we consider Rendezvous algorithms that can solve Rendezvous even from initial settings in which all robots have the same color. An algorithm solving Rendezvous is said to be *quasi-self-stabilizing* if it assumes that both robots always start with the same initial color chosen arbitrarily, and it is *self-stabilizing* if the robots can start from arbitrary colors.

A particular class of algorithms, denoted by $\mathcal{L}$, requires that robots only compute a destination point of the form $(1-\lambda)\cdot \text{me.position} + \lambda \cdot \text{other.position}$ for some $\lambda \in \mathbb{R}$, obtained as a function having only the colors as input (i.e., color of the other robot in the external-light) [21]. We call an algorithm in this class an $\mathcal{L}$-algorithm.

3 Previous Results for Rendezvous

Rendezvous is trivially solvable in FSYNC but is not in SSYNC in general.

▶ **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 as shown in the following theorems (or Table 1).

▶ **Theorem 2.** *Rendezvous is solved by self-stabilizing $\mathcal{L}$-algorithms under the following assumptions;*

1. *full-light with* 2 *colors, Non-Rigid and LC-atomic ASYNC* [17]*,*
2. *full-light with* 3 *colors, Non-Rigid and ASYNC* [21]*,*
3. *external-light with* 3 *colors, Non-Rigid and SSYNC* [10].

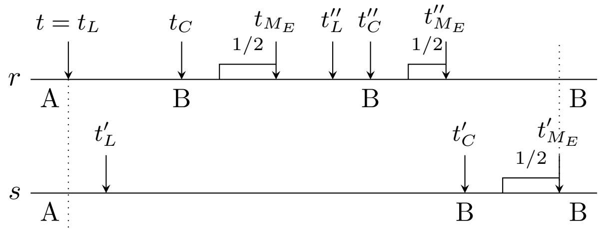


Figure 1 Move-atomic and LC-1-Bounded ASYNC schedule Rendezvous never succeeds.

▶ **Theorem 3.** [11] *Rendezvous is solved by a self-stabilizing non-$\mathcal{L}$-algorithm in full-light with* 2 *colors, Non-Rigid and ASYNC.*

▶ **Theorem 4.** [10] *Rendezvous is solved by non-quasi-self-stabilizing non-$\mathcal{L}$-algorithms under the following assumptions;*
1. *external-light with* 3 *colors, Non-Rigid($+\delta$) and ASYNC,*
2. *external-light with* 12 *colors, Rigid and ASYNC,*
3. *internal-light with* 3 *colors, Non-Rigid($+\delta$) and SSYNC,*
4. *internal-light with* 6 *colors, Rigid and SSYNC.*

Impossibility of Rendezvous $\mathcal{L}$-algorithms is stated as follows.

▶ **Theorem 5.**
1. *In ASYNC and Rigid, Rendezvous is not solvable by any quasi-self-stabilizing $\mathcal{L}$-algorithm with full-light of* 2 *colors* [21]*.*
2. *In ASYNC and Non-Rigid, Rendezvous is not solvable by any $\mathcal{L}$-algorithm with full-light of* 2 *colors* [21]*.*
3. *In Move-atomic but non-LC-atomic ASYNC and Rigid, Rendezvous is not solvable by any $\mathcal{L}$-algorithm with external-light of any number of colors* [10]*.*
4. *In SSYNC and Rigid, Rendezvous is not solvable by any $\mathcal{L}$-algorithm with internal-light of any number of colors* [10]*.*

Theorem 5 point **3** can be extended as follows;

▶ **Theorem 6.** *In Move-atomic and LC-*1*-Bounded ASYNC, and Rigid, Rendezvous is not solvable by any $\mathcal{L}$-algorithm with external-light of any number of colors.*

Proof. For each robot, the destination point and the next color are a function of the color of the other robot only. Assume that both robots start in the same color (say, A) and perform their execution synchronously. Consider a time t when both robots compute the midpoint m as a result of looking each other color A. Only robot r is let begin the cycle and perform *Look* operation, *Compute* one and M_E one at $t_L = t$, t_C and t_{M_E}, respectively. Robot r computes the midpoint m and changes its color to say, B at t_C. Robot s is let perform *Look* operation at time t'_L ($t_L < t'_L < t_C$), compute the midpoint m and change its color to B at time t'_C ($t_{M_E} < t'_C$), and move to m at time t'_{M_E} (Figure 1). Robot r is let end the next cycle before t'_C and perform *Look* operation, *Compute* one and M_E one at t''_L, t''_C and t'_{M_E}, respectively, where $t''_{M_E} < t'_C$. Since r keeps seeing s set to A in this cycle, r computes the new midpoint m' and changes its color to B. Then, both robots have the same color and does not attain Rendezvous at the time t'_{M_E}. By repeating the pattern, the robots never attain Rendezvous. Also this pattern satisfies *Move*-atomic and *LC*-1-Bounded ASYNC, and Rigid. ◀

Algorithm 1 SS-Rendezvous-with-3-colors (scheduler, movement, initial-color)[10].

Parameters: scheduler, movement-restriction, initial-color
Assumptions: external-light, three colors (A, B and C)

```
case other.light of
  A:
    me.light ← B
    me.des ← the midpoint of me.position and other.position
  B:
    me.light ← C
  C:
    me.light ← A
    me.des ← other.position
endcase
```

In the following sections, we consider $\mathcal{L}$-algorithms to solve Rendezvous on robots with external-lights and clarify the relationship among synchrony, the number of colors, movement restriction, and initial configurations.

4 Rendezvous $\mathcal{L}$-Algorithms for Robots with Three Colors of External Lights

In what follows, two robots are denoted as r and s. Let t_0 be the starting time of the algorithm. Given a robot *robot*, an operation $op(\in \{Look, Comp, LC, M_B, M_E\})$, and a time t, $t^+(robot, op)$ denotes the time *robot* performs the first *op* after t (inclusive) if there exists such operation, and $t^-(robot, op)$ denotes the time *robot* performs the first *op* before t (inclusive) 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* (*cs-time*, for short), if the next performed operations of both r and s after t_c are both *Look*, 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.

In [10], a Rendezvous algorithm is shown in SSYNC and Non-Rigid with external-light of three colors (Algorithm 1).

▶ **Theorem 7.** [10] *Rendezvous is solved by SS-Rendezvous-with-3-colors(SSYNC, Non-Rigid, any). It is a self-stabilizing $\mathcal{L}$-algorithm.*

We will show that Algorithm 1 does not work in even LC-atomic and *Move*-atomic ASYNC and Rigid, starting from the initial color A. In fact, in the next section, more generally we will show that there exist no $\mathcal{L}$-algorithms to solve Rendezvous in LC-atomic and *Move*-atomic ASYNC and Non-Rigid with three colors of external-lights. We also show that there exist no quasi-self-stabilizing $\mathcal{L}$-algorithms to solve Rendezvous if we change the assumption of Non-Rigid to Rigid. On the other hand, we show that there exists a non-quasi-self-stabilizing $\mathcal{L}$-algorithm to solve Rendezvous in LC-atomic ASYNC and Rigid with three colors of external-lights (Algorithm 2).

▶ **Theorem 8.** *Rendezvous is solved by NonQSS-Rendezvous-with-3-colors (LC-atomic ASYNC, Rigid, A). It is a non-quasi-self-stabilizing $\mathcal{L}$-algorithm.*

Algorithm 2 NonQSS-Rendezvous-with-3-colors (scheduler, movement, initial-color).

Parameters: scheduler, movement-restriction, initial-color
Assumptions: external-light, three colors (A, B and C)

```
case other.light of
  A:
    me.light ← B
    me.des ← the midpoint of me.position and other.position
  B:
    me.light ← C
  C:
    me.light ← B
    me.des ← other.position
endcase
```

(a) Algorithm 1 **(b)** Algorithm 2

Figure 2 Graph representations for Algorithms 1 (a) and 2 (b).

In the following, we derive lower bounds on the number of colors of external-lights. In order to do so, we introduce some notation for $\mathcal{L}$-algorithms and show their properties.

In $\mathcal{L}$-algorithms, the next color and destination (denoted as λ) is determined only by the current color observed by the robot. Thus an $\mathcal{L}$-algorithm is represented by an edge-labeled directed graph $G_{\mathcal{L}} = (V_{\mathcal{L}}, E_{\mathcal{L}}, \ell_{\mathcal{L}})$, where $V_{\mathcal{L}}$ is a set of colors used in the algorithm, $E_{\mathcal{L}}$ is a set of transitions from current colors observed by the robots to the next colors computed by the robots, and $\ell_{\mathcal{L}}$ is an edge-labeled function from $E_{\mathcal{L}}$ to $\mathbb{R}$. Edge $e = (c_1, c_2) \in E_{\mathcal{L}}$ and $\ell_{\mathcal{L}}(e) = \lambda$ mean that when a robot observes color c_1 of the other robot, it changes its color to c_2 and moves to the point decided by the value λ.[3] Also the out-degree of each node must be one, since we consider deterministic $\mathcal{L}$-algorithms. Thus, when the number of nodes in $G_{\mathcal{L}}$ is k, $G_{\mathcal{L}}$ has k edges. For example, Algorithms 1 and 2 are represented by the following directed graphs $G_{\mathcal{L}1}$ and $G_{\mathcal{L}2}$, respectively.

$G_{\mathcal{L}1} = (V_{\mathcal{L}1}, E_{\mathcal{L}1}, \ell_{\mathcal{L}1})$, where $V_{\mathcal{L}1} = \{A, B, C\}$, $E_{\mathcal{L}1} = \{(A, B), (B, C), (C, A)\}$ and $\ell_{\mathcal{L}1}((A, B)) = 1/2$, $\ell_{\mathcal{L}1}((B, C)) = 0$ and $\ell_{\mathcal{L}1}((C, A)) = 1$ (Figure 2(a)).

$G_{\mathcal{L}2} = (V_{\mathcal{L}2}, E_{\mathcal{L}2}, \ell_{\mathcal{L}2})$, where $V_{\mathcal{L}2} = \{A, B, C\}$, $E_{\mathcal{L}2} = \{(A, B), (B, C), (C, B)\}$ and $\ell_{\mathcal{L}2}((A, B)) = 1/2$, $\ell_{\mathcal{L}1}((B, C)) = 0$ and $\ell_{\mathcal{L}2}((C, B)) = 1$ (Figure 2(b)).

In what follows, we identify an $\mathcal{L}$-algorithm with its graph representation and $e = (c_1, c_2) \in E_{\mathcal{L}}$ and $\ell_{\mathcal{L}}(e) = \lambda$ are denoted as $c_1 \xrightarrow{\lambda} c_2$.

▶ **Lemma 9.** *Let $A_{\mathcal{L}}$ be an $\mathcal{L}$-algorithm solving Rendezvous in SSYNC and Rigid with external-light. If $A_{\mathcal{L}}$ starts from an initial setting such that both robots have the same color, then $A_{\mathcal{L}}$ has the following properties.*

[3] Note that $G_{\mathcal{L}}$ is not a state-transition graph.

1. *There is a color X such that $A_{\mathcal{L}}$ must have an edge $X \stackrel{1/2}{\to} Y$.*
2. *There is a color X such that $A_{\mathcal{L}}$ must have an edge $X \stackrel{1}{\to} Y$.*
3. *There is a color X such that $A_{\mathcal{L}}$ must have an edge $X \stackrel{0}{\to} Y$.*

Lemma 9 implies that any $\mathcal{L}$-algorithm must contain three different edges beginning with different colors.

▶ **Theorem 10.** [4] *In any Rendezvous $\mathcal{L}$-algorithm with external-light, robots must have three colors in SSYNC and Rigid.*

This theorem implies that Algorithm 1 has the optimal number of colors of external-lights in SSYNC. Note that it is self-stabilizing and works in Non-Rigid. On the other hand, if we assume Rigid movement, we can show the $\mathcal{L}$-algorithm with three colors to solve Rendezvous in LC-atomic ASYNC, which is however not quasi-self-stabilizing. In the next section, we will show a quasi-self-stabilizing $\mathcal{L}$-algorithm with four colors and a self-stabilizing $\mathcal{L}$-algorithm with five colors to solve Rendezvous in LC-atomic ASYNC and Non-Rigid. We will also show that the number of colors used in each algorithm is optimal.

5 Optimal Rendezvous $\mathcal{L}$-Algorithms for LC-atomic ASYNC Robots with External Lights

5.1 Lower Bounds

In this subsection, we first show that there exist no Rendezvous $\mathcal{L}$-algorithms with external light of 3 colors in LC-atomic and $Move$-atomic ASYNC in Non-Rigid.

If there exists such an $\mathcal{L}$-algorithm, the algorithm must be an edge-labeled directed graph $G_{\mathcal{L}} = (V_{\mathcal{L}}, E_{\mathcal{L}}, \ell_{\mathcal{L}})$ such that $V_{\mathcal{L}} = \{A, B, C\}$(three colors) and $\ell_{\mathcal{L}}(E_{\mathcal{L}}) = \{0, 1/2, 1\}$ (by Lemma9) and one of the following edge sets:
1. $E_{\mathcal{L}}$ contains a self-loop edge, say (A, A), and does not contain both directed edges,
2. $E_{\mathcal{L}}$ contains both directed edges, say (B, C) and (C, B), or
3. $E_{\mathcal{L}} = \{(A, B), (B, C), (C, A)\}$.

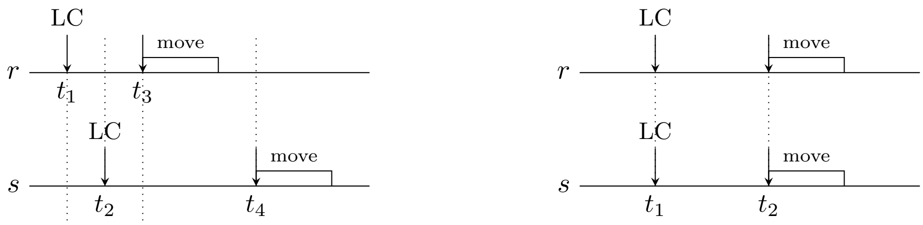
For Case 1. If the algorithm does not contain both directed edges, it can be verified that no algorithm can solve Rendezvous in SSYNC and Rigid. That is, if the algorithm starts with a color consisting of a self-loop edge, then it cannot solve Rendezvous since it cannot use more than one color. If the algorithm starts with a color not consisting of a self-loop edge, the color of both robots can be changed into the color with the self-loop edge without attaining Rendezvous. Thus, the algorithm also fails to Rendezvous in this case.

For Case 2. If algorithms do not contain self-loop edges, their graphs are the same as that of Algorithm 2. But it can be verified that Algorithm 2 fails to solve Rendezvous in SSYNC, Rigid and starting from color B or C, or SSYNC, Non-Rigid and starting from any color. It is easily verified that other algorithms with different edge-labeled functions fail to solve Rendezvous in SSYNC and Rigid starting from any color. If algorithms contain self-loop edges (both directed edges and a self-loop edge), since they can use only less than three colors even if starting from any color, they never solve Rendezvous in SSYNC and Rigid.

In Case 3, there are essentially two algorithms.

(a) $\ell_{\mathcal{L}}((A, B)) = 1/2$, $\ell_{\mathcal{L}}((B, C)) = 0$, and $\ell_{\mathcal{L}}((C, A)) = 1$ (denoted as Alg-(a)),
(b) $\ell_{\mathcal{L}}((A, B)) = 1/2$, $\ell_{\mathcal{L}}((B, C)) = 1$, and $\ell_{\mathcal{L}}((C, A)) = 0$ (denoted as Alg-(b)).

[4] This result is stated in [10, 21] but is not proved yet.



(a) Alternating (*alt*) **(b)** Simultaneous (*sim*)

Figure 3 Special schedules *alt* and *sim*.

Note that Alg-(a) is Algorithm 1.

We introduce special schedules to analyze $\mathcal{L}$-algorithms solving Rendezvous in LC-atomic ASYNC, with which we show that these algorithms do not work well.

Let $([\alpha_1, \beta_1], [\alpha_2, \beta_2], \ldots)$ be a sequence of operations that robots r and s perform, where r and s perform α_i and β_i at time t_i $(1 \leq i)$, respectively, and α_i and β_i are taken from LC-operation (denoted as LC), $Move$-operations, M_B, M_E or M (if $Move$-atomic) (denoted as M), and a "*no-op*" operation (denoted as $-$). For example, $([LC, -], [-, LC], [M, -], [-, M])$ denotes that r performs LC and M at times t_1 and t_3 and s performs LC and M at times t_2 and t_4, which is in LC-atomic $Move$-atomic ASYNC. Similarly, $([LC, LC], [M, M])$ denotes that r and s perform LC at time t_1 and perform M at time t_2, which is in FSYNC. The former is called alternating schedule and denoted as *alt* and the latter is called simultaneous schedule and denoted as *sim* (Figure 3).

Assume that r and s have colors c_r and c_s at some time t and let $d_t = dis(p(r,t), p(s,t))$. Let $(c_r, c_s; d_t)$ denote a configuration of a pair of colors of robots and its distance at t. When a configuration $(c_r, c_s; d_t)$ is changed into another one $(c'_r, c'_s; d_{t'})$ by performing an algorithm alg with a schedule sch, we denote $(c_r, c_s; d_t) \stackrel{sch}{\to} (c'_r, c'_s; d_{t'})_{alg}$, where t' is the time after which the robots have performed alg with the schedule sch. The suffix alg is usually omitted when the algorithm is apparent from the context.

We show that Alg-(a) and Alg-(b) cannot work from any initial configuration of the same color by using the schedules *alt* and *sim*.

▶ **Lemma 11.** *Alg-(a) cannot solve Rendezvous in LC-atomic and Move-atomic ASYNC and Rigid.*

▶ **Lemma 12.** *Alg-(b) cannot solve Rendezvous in LC-atomic and Move-atomic ASYNC and Rigid.*

▶ **Theorem 13.** *There exist no $\mathcal{L}$-algorithms of Rendezvous with external light of 3 colors in LC-atomic and Move-atomic ASYNC and Non-Rigid. Furthermore, there exist no quasi-self-stabilizing $\mathcal{L}$-algorithms of Rendezvous with external light of 3 colors in LC-atomic and Move-atomic ASYNC and Rigid.*

In an argument similar to the one above, we show that there exist no self-stabilizing $\mathcal{L}$-algorithms of Rendezvous with external-light of 4 colors in LC-atomic and $Move$-atomic ASYNC and Rigid.

If there exists such an $\mathcal{L}$-algorithm, the algorithm must be an edge-labeled directed graph $G_\mathcal{L} = (V_\mathcal{L}, E_\mathcal{L}, \ell_\mathcal{L})$ such that $V_\mathcal{L} = \{A, B, C, D\}$(four colors) and $\ell_\mathcal{L}(E_\mathcal{L}) \supseteq \{0, 1/2, 1\}$ (by Lemma 9). If the number of strongly connected components for $G_\mathcal{L}$ is at least two, then there exists an initial configuration of both robots with a same color, from which an algorithm

Algorithm 3 QSS-Rendezvous-with-4-colors (LC-atomic ASYNC, Non-Rigid, initial-color).

Parameters: scheduler, movement-restriction, initial-color
Assumptions: external-light, four colors (A, B, C and D)

```
case other.light of
A:
    me.light ← B
    me.des ← the midpoint of me.position and other.position
B:
    me.light ← C
C:
    me.light ← D
    me.des ← other.position
D:
    me.light ← A
endcase
```

cannot use four colors, it cannot solve Rendezvous by Theorem 13. Then the remaining case is that these graphs have one strongly connected component (one cycle) and have one of the following edge sets:

(1) $\ell_{\mathcal{L}}((A,B)) = 1/2$, $\ell_{\mathcal{L}}((B,C)) = 0$, $\ell_{\mathcal{L}}((C,D)) = 1$, and $\ell_{\mathcal{L}}((D,A)) = \lambda$(denoted as Alg-(1)),
(2) $\ell_{\mathcal{L}}((A,B)) = 1/2$, $\ell_{\mathcal{L}}((B,C)) = 1$, $\ell_{\mathcal{L}}((C,D)) = 0$, and $\ell_{\mathcal{L}}((D,A)) = \lambda$(denoted as Alg-(2)),
(3) $\ell_{\mathcal{L}}((A,B)) = 1/2$, $\ell_{\mathcal{L}}((B,C)) = 0$, $\ell_{\mathcal{L}}((C,D)) = \lambda$, and $\ell_{\mathcal{L}}((D,A)) = 1(\lambda \neq 1)$(denoted as Alg-(3)),
(4) $\ell_{\mathcal{L}}((A,B)) = 1/2$, $\ell_{\mathcal{L}}((B,C)) = 1$, $\ell_{\mathcal{L}}((C,D)) = \lambda$, and $\ell_{\mathcal{L}}((D,A)) = 0(\lambda \neq 0)$(denoted as Alg-(4)),
(5) $\ell_{\mathcal{L}}((A,B)) = 1/2$, $\ell_{\mathcal{L}}((B,C)) = \lambda$, $\ell_{\mathcal{L}}((C,D)) = 0$, and $\ell_{\mathcal{L}}((D,A)) = 1(\lambda \neq 1)$(denoted as Alg-(5)),
(6) $\ell_{\mathcal{L}}((A,B)) = 1/2$, $\ell_{\mathcal{L}}((B,C)) = \lambda$, $\ell_{\mathcal{L}}((C,D)) = 1$, and $\ell_{\mathcal{L}}((D,A)) = 0(\lambda \neq 0)$(denoted as Alg-(6)).

▶ **Lemma 14.** *Alg-(1)-Alg-(6) cannot solve Rendezvous in LC-atomic and Move-atomic ASYNC and Rigid from some initial configuration.*

▶ **Theorem 15.** *There exist no self-stabilizing $\mathcal{L}$-algorithms of Rendezvous with external-light of* 4 *colors in LC-atomic and Move-atomic ASYNC and Rigid.*

5.2 Optimal $\mathcal{L}$-algorithms

In this subsection, we show two optimal $\mathcal{L}$-algorithms of Rendezvous, one is quasi-self-stabilizing with 4 colors (Algorithm 3) and the other is self-stabilizing with 5 colors (Algorithm 4).

Algorithm 3 (QSS-Rendezvous-with-4-colors (LC-atomic ASYNC, Non-Rigid, initial-light)) satisfies the following lemmas. Let t_c be a cs-time of Algorithm 3.

The correctness proof proceeds as follows;

1. First we prove that Algorithm 3 is quasi-self-stabilizing. Algorithm 3 does not work from the initial configuration $\{\ell(r,t_0), \ell(s,t_0)\} = \{A, C\}$ or $\{\ell(r,t_0), \ell(s,t_0)\} = \{B, D\}$ (Lemma 14). However, we show that these configurations can not be reached from the initial configuration that r and s have a same color as follows. Assume that robots r

Algorithm 4 SS-Rendezvous-with-5-colors (LC-atomic ASYNC, Non-Rigid, initial-color).

Parameters: scheduler, movement-restriction, Initial-color)
Assumptions: external-light, five colors (A, B, C, D and E)

case other.light **of**
  A:
    $me.light \leftarrow B$
    $me.des \leftarrow$ the midpoint of $me.position$ and $other.position$
  B:
    $me.light \leftarrow C$
  C:
    $me.light \leftarrow D$
    $me.des \leftarrow other.position$
  D:
    $me.light \leftarrow E$
  E:
    $me.light \leftarrow A$
endcase

(a) Algorithm 3 **(b)** Algorithm 4

Figure 4 Graph representations for Algorithms 3 (a) and 4 (b).

and s start with a same color. If all LC-operations of r and s are performed at the same times, Rendezvous succeeds preserving that $\ell(r,t) = \ell(s,t)$ for any cs-time t (Lemma 16). Otherwise, there are different times at which LC-operations of r and s are performed and let t_r and t_s be the first times r and s are performed LC-operations, respectively ($t_r \neq t_s$). Then we show that there are colors α and β such that $\{\ell(r,t^*), \ell(s,t^*)\} = \{\alpha, \beta\}$ and $\alpha \to \beta$ for any cs-time t^* after the time $max(t_r, t_s)$ (Lemma 17). When robots r and s start with $\{\ell(r,t_c), \ell(s,t_c)\} = \{\alpha, \beta\}$ such that $\alpha \to \beta$, this relation of colors is preserved for any cs-time after t_c (Lemma 18).

2. Next we show that if robots r and s start with colors α and β such that $\alpha \to \beta$, Algorithm 3 attains Rendezvous. If $\{\alpha, \beta\} = \{B, C\}$, Rendezvous succeeds, or there is a cs-time $t^*(\geq t_c)$ such that the distance between r and s at t^* decreases at least δ from the distance at t_c and $\{\ell(r,t^*), \ell(s,t^*)\} = \{C, D\}$ or $\{\ell(r,t^*), \ell(s,t^*)\} = \{D, A\}$ (Lemma 22). If $\{\alpha, \beta\} = \{A, B\}, \{C, D\}$, or $\{D, A\}$, then there is a cs-time $t^*(\geq t_c)$ such that the distance between r and s at t^* is less than or equal to the distance at t_c and $\{\ell(r,t^*), \ell(s,t^*)\} = \{B, C\}$ (Lemma 23).

▶ **Lemma 16.** *Let $\ell(r,t_c) = \ell(s,t_c)$. Assume that all LC-operations of r and s are performed at the same times, and let $t_i (i = 1, 2, 3, \dots)$ be the times r and s perform LC-operations simultaneously. Then $\ell(r,t_i) = \ell(s,t_i)$ for any time $t_i (i = 1, 2, 3, \dots)$ and there is a cs-time $t^*(\geq t_c)$ such that $dis(p(r,t^*), p(s,t^*)) = 0$.*

▶ **Lemma 17.** *Let $\ell(r,t_c) = \ell(s,t_c)$. Assume that there are different times at which LC-operations of r and s are performed and let t_r and t_s be the first times r and s are performed LC-operations, respectively ($t_r \neq t_s$). Then there are colors α^* and β^* such that $\{\ell(r,t^*), \ell(s,t^*)\} = \{\alpha^*, \beta^*\}$ and $\alpha^* \to \beta^*$ for the first cs-time t^* after $max(t_r, t_s)$.*

The following lemma can be proved similar to Lemmas 16-17.

▶ **Lemma 18.** *Assume that Algorithm 3 starts with $\{\ell(r,t_c), \ell(s,t_c)\} = \{\alpha, \beta\}$ such that $\alpha \to \beta$. Let t^* be the first cs-time after t_c. Then there are colors α^* and β^* such that $\{\ell(r,t^*), \ell(s,t^*)\} = \{\alpha^*, \beta^*\}$ and $\alpha^* \to \beta^*$.*

▶ **Lemma 19.** *Let $\ell(r,t_c) = \ell(s,t_c)$. If Algorithm 3 is performed starting from t_c, there does not exist any cs-time $t^*(\geq t_c)$ such that $\{\ell(r,t^*), \ell(s,t^*)\} = \{A, C\}$ or $\{\ell(r,t^*), \ell(s,t^*)\} = \{B, D\}$.*

Proof. Configuration of $\{\ell(r,t^*), \ell(s,t^*)\} = \{A, C\}$ or $\{\ell(r,t^*), \ell(s,t^*)\} = \{B, D\}$ at any cs-time t^* cannot be reached from any initial configuration that r and s have a same color by Lemmas 16-18. ◀

▶ **Lemma 20.** *If $dis(p(r,t_c), p(s,t_c)) = 0$ and Algorithm 3 is performed starting from t_c, $dis(p(r,t), p(s,t)) = 0$ for any $t \geq t_c$.*

Proof. Since $dis(p(r,t_c), p(s,t_c)) = 0$, any move operation becomes no move (stay). ◀

▶ **Lemma 21.** *Let $\alpha = B$ and $\beta = C$ or $\alpha = D$ and $\beta = A$. If $\ell(r,t_c) = \alpha$ and $\ell(s,t_c) = \beta$ in Algorithm 3, then $\ell(s,t) = \beta$ and $p(s,t) = p(s,t_c)$ for any $t(t_c \leq t \leq t_1 = t_c^+(r, LC))$.*

Proof. When s with color β observes r with color α at $t(t_c \leq t \leq t_1)$, s does not change its color at time t and stays at position $p(s,t_c)$. ◀

▶ **Lemma 22.** *If Algorithm 3 starts with $\{\ell(r,t_c), \ell(s,t_c)\} = \{B, C\}$, for any schedule of two robots after t_c, there is a cs-time $t^*(\geq t_c)$ such that $dis(p(r,t^*), p(s,t^*)) = 0$, or $dis(p(r,t^*), p(s,t^*)) \leq dis(p(r,t_c), p(s,t_c)) - \delta$ and $\{\ell(r,t^*), \ell(s,t^*)\} = \{C, D\}$ or $\{\ell(r,t^*),\ \ell(s,t^*)\} = \{D, A\}$.*

▶ **Lemma 23.** *If Algorithm 3 starts with $\{\ell(r,t_c), \ell(s,t_c)\} = \{\alpha, \beta\}$ such that $\alpha \to \beta$, for any schedule of two robots after t_c, there is a cs-time $t^*(\geq t_c)$ such that $\{\ell(r,t^*), \ell(s,t_c)\} = \{B, C\}$ and $dis(p(r,t^*), p(s,t^*)) \leq dis(p(r,t_c), p(s,t_c))$.*

Lemmas 16-23 follow the next theorem.

▶ **Theorem 24.** *Rendezvous is solved by QSS-Rendezvous-with-4-colors(LC-atomic ASYNC, Non-Rigid, any) with $\ell(r,t_0) = \ell(s,t_0)$. It is a quasi-self-stabilizing $\mathcal{L}$-algorithm.*

Algorithm 4 also satisfies similar properties of Lemmas 16-23 (Lemmas 26-33) and it can be also shown to be a self-stabilizing $\mathcal{L}$-algorithm by using these lemmas and the following Lemma 25. In Algorithm 3, two color pairs $\{A, C\}$ and $\{B, D\}$ of r and s cannot be reached from any initial configuration with same colors (Lemma 19). However, it cannot achieve Rendezvous from the initial configuration $\{A, C\}$ or $\{B, D\}$ (Lemma 14), since repetitions of $\{A, C\}$ and $\{B, D\}$ never attain Rendezvous. This is the reason why Algorithm 3 is not self-stabilizing. On the other hand, we can show that Algorithm 4 is self-stabilizing. In fact, it can solve Rendezvous from the initial configurations $\{A, C\}$, $\{B, D\}$, $\{C, E\}$, $\{D, A\}$ or $\{E, B\}$ as expressed in the following Lemma 25. Even if these configurations repeat, since the repetition contains $\{C, E\}$, Rendezvous succeeds. Otherwise, any configuration can reach some configuration $\{\alpha, \beta\}$ ($\alpha \to \beta$).

▶ **Lemma 25.** *Let $\{\ell(r,t_c),\ell(s,t_c)\}=\{\alpha,\gamma\}$, where $\alpha\to\beta$ and $\beta\to\gamma$. If Algorithm 4 starts with from any configuration $\{\ell(r,t_c),\ell(s,t_c)\}=\{\alpha,\gamma\}$, for any schedule of two robots after t_c, there is a cs-time $t^*(\geq t_c)$ such that*
$dis(p(r,t^*),p(s,t^*))=0$, or $dis(p(r,t^*),p(s,t^*))\leq dis(p(r,t_c),p(s,t_c))-\delta$
and $\{\ell(r,t^*),\ell(s,t^*)\}=\{\alpha',\beta'\}$ for some α' and β' $(\alpha'\to\beta')$.

The remaining lemmas can be proved similar to Lemmas 16-23.

▶ **Lemma 26.** *Let $\ell(r,t_c)=\ell(s,t_c)$ in Algorithm 4. If all LC-operations of r and s are performed at the same times, and let $t_i(i=1,2,3,\ldots)$ be the times r and s perform LC-operations simultaneously. Then $\ell(r,t_i)=\ell(s,t_i)$ for any time $t_i(i=1,2,3,\ldots)$ and there is a cs-time $t^*(\geq t_c)$ such that $dis(p(r,t^*),p(s,t^*))=0$.*

▶ **Lemma 27.** *Let $\ell(r,t_c)=\ell(s,t_c)$ in Algorithm 4. Assume that there are different times at which LC-operations of r and s are performed. Let t^* be the first cs-time after the first different time at which different LC-operations of r and s are performed. Then there are colors α^* and β^* such that $\{\ell(r,t^*),\ell(s,t^*)\}=\{\alpha^*,\beta^*\}$ and $\alpha^*\to\beta^*$.*

▶ **Lemma 28.** *Assume that Algorithm 4 starts with $\{\ell(r,t_c),\ell(s,t_c)\}=\{\alpha,\beta\}$ such that $\alpha\to\beta$. Let t^* be the first cs-time after t_c. Then there are colors α^* and β^* such that $\{\ell(r,t^*),\ell(s,t^*)\}=\{\alpha^*,\beta^*\}$ and $\alpha^*\to\beta^*$.*

▶ **Lemma 29.** *Let $\ell(r,t_c)=\ell(s,t_c)$. If Algorithm 4 is performed starting from t_c, there exist no cs-time $t^*(\geq t_c)$ such that $\{\ell(r,t^*),\ell(s,t^*)\}=\{\alpha,\gamma\}$, where $\alpha\to\beta$ and $\beta\to\gamma$.*

▶ **Lemma 30.** *If $dis(p(r,t_c),p(s,t_c))=0$ and Algorithm 4 is executed starting from t_c, $dis(p(r,t),p(s,t))=0$ for any $t\geq t_c$.*

▶ **Lemma 31.** *Let $\alpha=B$ and $\beta=C$, $\alpha=D$ and $\beta=E$, or $\alpha=E$ and $\beta=A$. If $\ell(r,t_c)=\alpha$ and $\ell(s,t_c)=\beta$ in Algorithm 4, then $\ell(s,t)=\beta$ and $p(s,t)=p(s,t_c)$ for any $t(t_c\leq t\leq t_c^+(r,LC))$.*

▶ **Lemma 32.** *If Algorithm 4 starts with $\{\ell(r,t_c),\ell(s,t_c)\}=\{B,C\}$, for any schedule of two robots after t_c, there is a cs-time $t^*(\geq t_c)$ such that $dis(p(r,t^*),p(s,t^*))=0$, or $dis(p(r,t^*),p(s,t^*))\leq dis(p(r,t_c),p(s,t_c))-\delta$ and $\{\ell(r,t^*),\ell(s,t^*)\}=\{C,D\}$ or $\{\ell(r,t^*),\ell(s,t^*)\}=\{D,A\}$.*

▶ **Lemma 33.** *If Algorithm 4 starts with $\{\ell(r,t_c),\ell(s,t_c)\}=\{\alpha,\beta\}$ such that $\alpha\to\beta$, for any schedule of two robots after t_c, there is a cs-time $t^*(\geq t_c)$ such that $\{\ell(r,t^*),\ell(s,t_c)\}=\{B,C\}$ and $dis(p(r,t^*),p(s,t^*))\leq dis(p(r,t_c),p(s,t_c))$.*

Lemmas 25-33 follow the next theorem.

▶ **Theorem 34.** *Rendezvous is solved by SS-Rendezvous-with-5-colors(LC-atomic ASYNC, Non-Rigid, any). It is a self-stabilizing $\mathcal{L}$-algorithm.*

6 Concluding Remarks

We have shown that Rendezvous can be solved by $\mathcal{L}$-algorithms in LC-atomic ASYNC with the optimal number of colors of external-lights in the following cases. (1) Rigid and non-quasi-self-stabilizing, (2) Non-Rigid and quasi-self-stabilizing, and (3) Non-Rigid and self-stabilizing. We have also shown impossibility result that Rendezvous cannot be solved by any $\mathcal{L}$-algorithm with any number of colors of external-lights in LC-1-Bounded ASYNC. Combining it with our algorithms in LC-atomic ASYNC, we have shown that LC-atomic ASYNC is the maximal subclass in ASYNC Rendezvous can be solved by $\mathcal{L}$-algorithms with external-lights.

References

1. N. Agmon and D. Peleg. Fault-tolerant gathering algorithms for autonomous mobile robots. *SIAM J. Comput.*, 36(1):56–82, 2006.
2. Z. Bouzid, S. Das, and S. Tixeuil. Gathering of mobile robots tolerating multiple crash Faults. In *33rd ICDCS*, pages 334–346, 2013.
3. M. Cieliebak, P. Flocchini, G. Prencipe, and N. Santoro. Distributed computing by mobile robots: Gathering. *sicomp*, 41(4):829–879, 2012.
4. S. Das, P. Flocchini, G. Prencipe, N. Santoro, and 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, and P. Raipin Parvédy. Fault and Byzantine tolerant self-stabilizing mobile robots gathering - Feasibility study. CoRR abs/1602.05546, arXiv, 2016.
6. B. Degener, B. Kempkes, T. Langner, F. Meyer auf der Heide, P. Pietrzyk, and R. Wattenhofer. A tight run-time bound for synchronous gathering of autonomous robots with limited visibility. In *23rd ACM SPAA*, pages 139–148, 2011.
7. Y. Dieudonné and F. Petit. Self-stabilizing gathering with strong multiplicity detection. *tcs*, 428(13):47–57, 2012.
8. P. Flocchini, G. Prencipe, and N. Santoro. *Distributed Computing by Oblivious Mobile Robots*. Morgan & Claypool, 2012.
9. P. Flocchini, G. Prencipe, N. Santoroand, and P. Widmayer. Gathering of asynchronous robots with limited visibility. *tcs*, 337(1–3):147–169, 2005.
10. P. Flocchini, N. Santoro, G. Viglietta, and M. Yamashita. Rendezvous with Constant Memory. *tcs*, 621:57–72, 2016.
11. A. Hériban, X. Défago, and S. Tixeuil. Optimally gathering two robots. In *19th ICDCN:3*, pages 1–10, 2018.
12. T. Izumi, Z. Bouzid, S. Tixeuil, and K. Wada. Brief Announcement: The BG-simulation for Byzantine mobile robots. In *25th DISC*, pages 330–331, 2011.
13. T. Izumi, Y. Katayama, N. Inuzuka, and K. Wada. Gathering Autonomous Mobile Robots with Dynamic Compasses: An Optimal Result. In *21st DISC*, pages 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. *sicomp*, 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. In *18th SIROCCO*, pages 150–161, 2011.
16. J. Lin, A.S. Morse, and B.D.O. Anderson. The multi-agent rendezvous problem. parts 1 and 2. *sicomp*, 46(6):2096–2147, 2007.
17. T. Okumura, K. Wada, and Y. Katayama. Brief Announcement: Optimal asynchronous Rendezvous for mobile robots with lights. In *19th SSS*, pages 484–488, 2017.
18. G. Prencipe. Impossibility of gathering by a set of autonomous mobile robots. *tcs*, 384(2–3):222–231, 2007.
19. 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.
20. I. Suzuki and M. Yamashita. Distributed anonymous mobile robots: Formation of geometric patterns. *sicomp*, 28:1347–1363, 1999.
21. G. Viglietta. Rendezvous of two robots with visible bits. In *ALGOSENSORS 2013*, pages 291–306, 2014.