Brief Announcement: Optimal Asynchronous Rendezvous for 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[4, 10]. 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[10], 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 always 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 non-rigid movement if we consider some reasonable restricted class of asynchronous settings.

1 Introduction

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, 2, 6, 9]. 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 execute identical algorithms)[3]. Robots operate in Look-Compute-Move (LCM) cycles in the model. In the Look operation, robots obtain a snapshot of the environment (locations of other robots) 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 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.

Previous Results In this paper, we focus on Rendezvous in asynchronous settings and we reveal the weakest additional assumptions for Rendezvous. 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 always 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.

Table 1. Rendezvous algorithms by robots with lights.

scheduler	movement	full-light[10]	external-light[4]	internal-light[4]	no-light[3, 8]
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.

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[10], with 3 colors in Non-Rigid($+\delta$) and external-light[4], and with 12 colors in Rigid and internal-light[4], respectively.

In this paper we consider Rendezvous algorithms in ASYNC with the optimal number of colors of 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 Non-Rigid.

2 Asynchronous Rendezvous Algorithms for Robots with Lights

The details of the model of autonomous mobile robots and necessary terminologies and all proofs are included in [7].

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 ← B
        me.des ← the midpoint of me.position and other.position
    else me.des ← other.position
B:
    if other.light = A then
        me.des ← me.position // stay
    else me.light ← 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[4].

Theorem 1. *Rendezvous(ASYNC, Rigid, A) solves Rendezvous. That is, Rendezvous can be solved in ASYNC and Rigid movement with* 2 *colors if the initial configuration is predetermined.*

[4] This algorithm is essentially the same as **Algorithm** 1 in [10].

LC-atomic ASYNC and Non-Rigid movement
Algorithm 1 belongs to the class $\mathcal{L}$ [10], where an algorithm is in $\mathcal{L}$ if every destination and every next color of light computed in the algorithm depend only on the current colors of the two robot's lights. It is shown in [10] that there is no algorithm of class $\mathcal{L}$ that solves Rendezvous using 2 colors, in ASYNC and Non-Rigid movement even assuming that both robots are set to a predetermined color in the initial configuration, and Rendezvous can be solved with an $\mathcal{L}$ algorithm using 3 colors in ASYNC and Non-Rigid movement regardless of the colors in the initial configuration. We show a sufficient condition of scheduler (LC-atomic ASYNC) in which Algorithm 1 (an $\mathcal{L}$ algorithm) solves Rendezvous with 2 colors in ASYNC and Non-Rigid movement from any initial configuration.

Theorem 2. *Rendezvous(LC-atomic ASYNC, Non-Rigid, any) solves Rendezvous. That is, Rendezvous can be solved by an $\mathcal{L}$-algorithm in LC-atomic ASYNC and Rigid movement with* 2 *colors regardless of the initial configuration.*

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).

Theorem 3. *RedezvousWithDelta(ASYNC, Non-Rigid($+\delta$), A) solves Rendezvous. That is, Rendezvous can be solved in ASYNC and Non-Rigid movement with* 2 *colors if robots know the value δ and the initial configuration is predetermined.*

3 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[5], 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.

[5] Very recently it has been solved affirmatively[5].

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

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

```
case dis(me.position, other.position)(= DIST) of
DIST > 2δ:
    if me.light =other.light =B then
        me.des ← the point moving by δ/2 from me.position to other.position
    else me.light ← B
2δ ≥ DIST ≥ δ:
    if me.light = other.light = A then
        me.light ← B
        me.des ← the midpoint of me.position and other.position
    else me.light ← A
δ > DIST: //Rendezvous(ASYNC, Rigid, A)
    case me.light of
    A:
        if other.light =A then
            me.light ← B
            me.des ← the midpoint of me.position and other.position
        else me.des ← other.position
    B:
        if other.light = A then me.des ← me.position // stay
        else me.light ← A
    endcase
endcase
```


2. 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.
3. P Flocchini, G Prencipe, N Santoro, Distributed Computing by Oblivious Mobile Robots, Synthesis Lectures on Distributed Computing Theory, Morgan & Claypool, 2012.
4. P.Flocchini , N.Santoro , G.Viglietta , M.Yamashita, Rendezvous with Constant Memory, Theoretical Computer Science, 621, 57–72, 2016.
5. A. Heriban, X. Défago, S. Tixeuil, Optimally gathering two robots, Research Report HAL Id: hal-01575451, UPMC Sorbonne Universités, Aug. 2017.
6. 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.
7. T. Okumura, K. Wada and Y. Katayama, Optimal asynchronous rendezvous for mobile robots with lights, Technical Report arXiv:1707.04449v1, July 2017.
8. G. Prencipe, Impossibility of gathering by a set of autonomous mobile robots, Theoretical Computer Science, 384(2-3):222–231, 2007.
9. I. Suzuki and M. Yamashita, Distributed anonymous mobile robots: Formation of geometric patterns, SIAM *Journal on Computing*, 28, 1347–1363, 1999.
10. G. Viglietta, Rendezvous of two robots with visible bits, Technical Report arXiv:1211.6039, 2012.