

# 非同期自律分散ロボット群に対する集合問題の可解性について

中井 陸雄<sup>1</sup> 和田幸一<sup>2</sup>

<sup>1</sup> 法政大学大学院理工学研究科 応用情報工学専攻

<sup>2</sup> 法政大学理工学部 応用情報工学科

## 1. はじめに

複数のコンピュータが通信を行い、協調して処理を行うシステムを分散システムと呼ぶ。この分散システムの分野の1つとして、複数のロボットがそれぞれで計算し、協調することで目的を達成する自律分散ロボット群と呼ばれるシステムがある。近年、この自律分散ロボット群の研究が盛んに行われている。

分散システムで考えられている環境は様々なクラスがあるが、自律分散ロボット群のロボットは互いに通信を行わず、自身で計算し移動を行うという特徴がある。そのため理論モデル[1]では、ロボットを平面上の動く点として扱う。また自律分散ロボット群のロボットは個体差を持たないため、同じアルゴリズムで動き、外見で個体を識別することができない。

ロボットは観測(Look)、計算(Compute)、移動(Move)を繰り返し行う。このサイクルをLCMサイクルと呼ぶ。Lookでは周囲の環境を観測し、ComputeではLookによって得た情報を元に目的地を計算する。MoveではComputeによって求めた目的地に移動する。

ロボットのスケジューラとして3つのモデルがある。FSYNC(完全同期)モデルではLCMサイクルの各命令を全てのロボットが同時に行う。SSYNC(準同期)モデルでは1台以上のロボットが同時に行う。ASYN(非同期)モデルではLCMサイクルが非同期的に行われる。

このロボットを利用して考えられている問題としてランデブー問題や集合問題、パターン形成などがある。本研究では集合問題についてのアルゴ

リズムを考案する。集合問題は、事前に知ることのできないある1点に全てのロボットを集める問題である。ロボットの台数が2台の場合の集合問題をランデブー問題と呼ぶ。また初期配置で全てのロボットが異なる点にいる集合問題はdistinct gatheringと呼ぶ。他にも移動距離が最小となる集合問題をmin-max gatheringと呼ぶ。集合問題は様々な仮定の下で、現在まで多くの研究が行われている[2-16]。

自律分散ロボット群のアルゴリズムの研究での目的は、問題を解決するために必要となる必要最低限の機能を明らかにすることである。ここで集合問題の可解性が変わる仮定について示す。まずMoveでのロボットの移動性として3つのモデルがある。まずnon-rigidは目的地にたどり着く保証がないが、少なくとも最小移動距離 $\delta > 0$ は移動する。次にnon-rigid(+ $\delta$ )ではnon-rigidの動きに加え、最小移動距離 $\delta$ を予め知っている。最後のrigidは必ず目的地にたどり着くことができる。

ロボットにライトと呼ばれる定数ビットの記憶領域を持つモデル[17]が提案されている。ライトの見え方には3種類ある。まずFull-Lightは自身のいる点と周囲の点の状態を観測することができる。次にExternal-Lightは周囲の点の状態を観測することができる。最後にInternal-Lightは自身のいる点の状態を観測することができる。

ロボットに多重検知を搭載したモデル[1]が提案されている。多重検知により、ロボットがある点に複数台存在するか判別することができる。多重検知できる範囲によって2種類に分けられる。周囲の点と自身のいる点が多重点か認識できるも

表 1 ライト付きモデルでの集合問題の従来の結果

スケジューラ	動き	full-light	external-light	internal-light	no-light
FSYNC					○
SSYNC	rigid		2*,3	?	×
	non-rigid	2	?	?	
	non-rigid(+ $\delta$ )		?	2**,3	
ASYNC	rigid	2(CM)	?	?	
	non-rigid	10	?	?	

\*局所多重検知, \*\*初期状態における 2 点の距離が  $2\delta$  以上

のを全体多重検知, 自身のいる点のみ多重点か認識できるものを局所多重検知と呼ぶ.

従来の研究によって, ライト付きモデルで考えられている集合問題を解決するアルゴリズムを表 1 に示す. また多重検知モデルでは, 以下の 2 つのアルゴリズムが提案されている.

- ① ロボットの台数が奇数の場合にのみ, SSYNC, 全体多重検知で集合問題を解くアルゴリズム [1]
- ② ロボットの台数が 3 台以上で ASYNC, 全体多重検知で distinct gathering を解くアルゴリズム [1]

本研究では ASYNC において, 基本モデルにできるだけ付加条件を抑えて集合問題を解くことを目標としている. 本論文ではロボットの台数が 3 台の場合に ASYNC, 局所多重検知, non-rigid で集合問題を解くアルゴリズムを提案する. また 4 台の場合に ASYNC, 局所多重検知, non-rigid, chirality で distinct gathering を解くアルゴリズムを提案する.

## 2. ロボットのモデル

自律分散ロボット群で扱うロボットはモデル化されており, ロボットはユークリッド平面  $\mathbb{R}^2$  の動く点として表される. システムを構成するロボットの集合  $\mathcal{R}$  は  $\mathcal{R} = \{r_1, \dots, r_n\}$  と表す. ある時刻  $t$  のロボットの位置は  $r(t)$  と表す. 曖昧でないならば, 時刻  $t$  を省略する場合がある. ロボット  $r_i$  は自身が原点となる局所座標系を持つ. 局所座標系での単位あたりの長さはロボットごとに異なる. また局

所座標系は X 軸と Y 軸の 2 つの座標軸を持つ直交座標系である. ロボット間での局所座標系の軸の合意には 4 つの種類がある [1].

- ConsistentCompass  
局所座標系の軸の向きと正負の方向が全てのロボットで一致している.
- OneAxis  
局所座標系の 1 つの軸の向きと正負の方向が全てのロボットで一致している.
- Chirality  
局所座標系の軸の向きが全てのロボットで一致している. そのため時計回りや反時計回りといった方向が一致する.
- Disorientation  
局所座標系の軸の向きと正負の方向が全てのロボットで一致していない.

またロボットは個体差がないため, 外見で識別することができず, 同じ機能を持ち, 同じアルゴリズムを実行する. またロボットは過去の記憶を持たないため, 過去の行動や情報を利用することができない. 過去の情報を必要としないため, 異常が起こった場合であっても計算を正しく行うことができるため故障に強い.

ロボットは他のロボットを観測(Look)し, 計算(Compute)し, 移動(Move)するという動作を繰り返し行う. これを LCM サイクルと呼ぶ.

### 1. Look

スナップショットを撮り, 可視範囲にあるロボットの座標の集合を得る.

## 2. Compute

Look で得た局所座標系からアルゴリズムに従い、自分の目的地を計算する。

## 3. Move

Compute で計算した目的地に向かって移動を行う。

Move でのロボットの動きとして3つのモデルがある。Compute で計算された目的地に必ず到達する場合を rigid といい、目的地に到達しない場合を non-rigid という。ただし、non-rigid の場合はある  $\delta > 0$  が存在して、少なくとも  $\delta$  は移動するものとする。目的地までの距離が  $\delta$  以下であれば必ず目的地までたどり着く、目的地までの距離が  $\delta$  より大きければ少なくとも  $\delta$  の位置まで移動する。また、non-rigid でロボットが  $\delta$  の値を知っている場合を non-rigid( $+\delta$ )と表す。

ロボットのスケジューリングとして基本的に3つのモデルがある。

- FSYNC(完全同期)

全てのロボットが同時刻に Look, Compute, Move を行う。

- SSYNC(準同期)

1台以上のロボットが起動され、起動したロボットは FSYNC と同様に同時刻に命令を行う。

- ASYNC(非同期)

ロボットが非同期的に起動される。このとき Look, Compute, Move にかかる時間は有限であるが、命令にかかる時間は予測することができない。そのため ASYNC では Look によってロボットが移動している最中に、他のロボットが観測する場合がある。

また ASYYNC モデルに制限を与えたクラスがある。[18,19]

- LC-Atomic

Look 命令と Compute 命令が同時刻に実行される。そのため Look 命令を実行してから Compute 命令で状態を変えた後、他のロボット

は状態を観測できる。

- Move-Atomic

Move 命令と Move 終了命令が同時刻に実行される。そのため瞬間的に移動していることと同義である。

- CM-Atomic

Compute 命令と Move 命令が同時刻に実行される。そのため瞬間的に移動した後他のロボットは状態を観測することとなる。

ロボットに追加の機能を持つモデルがある。その1つとして多重検知を持つモデルがある。この多重検知により、ロボットがある点に複数台存在するか判別することができる。複数台のロボットがいる点のことを多重点、多重点にいるロボットの台数のことを多重度と呼ぶ。多重検知できる範囲に2種類ある。周囲の点と自身のいる点が多重点か認識できるものを全体多重検知(global multiplicity detection)、自身のいる点が多重点か認識できるものを局所多重検知(local multiplicity detection)と呼ぶ。さらにこの多重検知には強さがあり、強い多重検知では多重点の多重度を知ることができる。弱い多重検知では多重点であることを知ることができるが、多重度は知ることができない。

他にもライトと呼ばれる定数ビットの記憶領域を持つモデルがある。ライトの状態は Compute 命令によって更新することができる。ライトの見え方には3種類ある。自身のいる点と周囲の点の状態を観測ができるものを Full-Light、周囲の点の状態のみを観測できるものを External-Light、自身のいる点の状態のみを観測できるものを Internal-Light と呼ぶ。またその点を観測した時の見え方にも種類がある。その点に存在する全てのロボットの状態を認識できるものを multiset-view、重複要素を除いた状態の集合として認識するものを set-view、その点に存在するロボットのうち任意の1つの状態を認識するものを arbitrary-view と呼ぶ。

### 3. 集合問題について

複数のロボットが任意の初期配置から、事前に知らないある1点に集合する問題を集合問題と呼ぶ。この集合問題では初期配置で多重点が生まれる場合がある。初期配置で多重点がない場合の集合問題を distinct gathering と呼ぶ。

### 4. 集合問題に対する従来の結果

集合問題は FSYNC モデルでは解くことができるが、SSYNC モデル、ASync モデルでは解くことができない。

定理 1. [1]

集合問題は SSYNC モデル、ASync モデルのロボットでは解くことができない。

多重検知モデルでの集合問題が可解となる条件を以下の定理で示す。

定理 2. [1]

- ① SSYNC, 全体多重検知で集合問題を解けるための必要十分条件は  $n$  が奇数であることである。
- ② ロボットの台数が3台以上で ASync, 全体多重検知で distinct gathering を解ける。

ライトモデルでの集合問題が可解となる条件を以下の定理で示す。

定理 3.

- ① SSYNC, full-light, non-rigid, 2 状態で集合問題を解ける。[20]
- ② ASync, full-light, non-rigid, 10 状態で集合問題を解ける。[20]
- ③ CM-Atomic ASync, full-light, rigid, 2 状態で集合問題を解ける。[18]
- ④ SSYNC, external-light, rigid, 局所多重検知, 2 状態で集合問題を解ける。[20]
- ⑤ SSYNC, external-light, rigid, 3 状態で集合問題を解ける。[20]
- ⑥ SSYNC, internal-light, non-rigid( $+\delta$ ), 初

期状況においてロボットが存在する点が2点であり、2点の距離が  $2\delta$  以上である場合に2状態で集合問題を解ける。[20]

- ⑦ SSYNC, internal-light, non-rigid( $+\delta$ ), 3 状態で集合問題を解ける。[18]

現在, ASync の多重検知モデルでは, 全体多重検知を用いた集合問題が考えられてきた。本研究では, より弱い仮定である局所多重検知を用いて集合問題を解くことを目標としている。

### 5. 本論文の結果

本論文では以下2つの定理を示す。

定理 4.

$n=3$  ならば ASync, 局所多重検知, non-rigid で集合問題が解ける

定理 5.

$n=4$  ならば ASync, 局所多重検知, non-rigid, chirality で distinct gathering が解ける

またアルゴリズムに関して 5.1, 5.2 で説明する。

#### 5.1. $n=3$ の場合の集合アルゴリズム

$n=3$  の場合の ASync, 局所多重検知, non-rigid で集合問題を解くアルゴリズムを Algorithm1 に示す。

アルゴリズムで用いる表記を以下の通りとする。

$SS_i$ : ロボット  $r_i$  のスナップショット

$P(SS_i)$ :  $SS_i$  に含まれるロボットの位置の集合

$p_f$ : フェルマー点

図1に  $120^\circ$  以上の内角が存在しない場合を示す。3点を  $p_0, p_1, p_2$  としたとき  $\angle(p_0, p_f, p_1) = \angle(p_1, p_f, p_2) = \angle(p_0, p_f, p_2) = 120^\circ$  となるフェルマー点  $p_f$  が存在する。

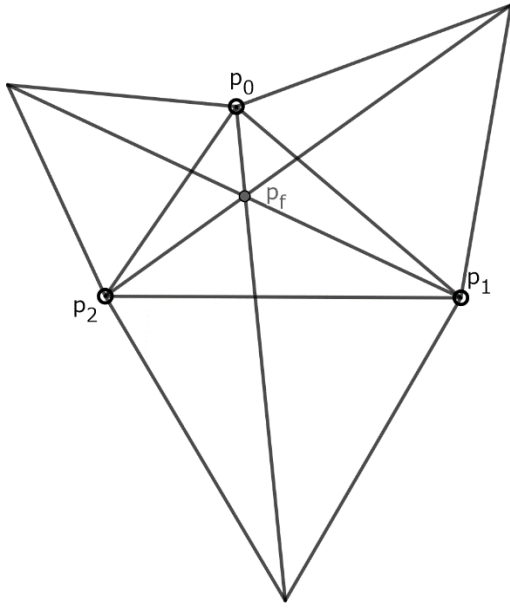


図 1 120° 以上の内角が存在しない場合

---

Algorithm1. Gathering 3

---

仮定 :  $n=3$ , ASYNC, 局所多重検知, non-rigid

- 1: If  $|P(SS_i)|=2$
- 2:   If  $r_i$ が多重点
- 3:     停止
- 4:   Else
- 5:     他方に向かって進む
- 6: If  $|P(SS_i)|=3$
- 7:   If 3点が一直線上に存在する
- 8:     (3点を結ぶ線分の端点を $p_e$ , 内点を $p_m$ とする)
- 9:     If  $r_i = p_e$
- 10:        $p_m$ へ移動
- 11:     Else
- 12:       何もしない
- 13:   Else
- 14:    $(P(SS_i)=\{p_0, p_1, p_2\},$   
 $\alpha_j = \angle(p_{j-1}, p_j, p_{j+1}) (0 \leq j \leq 2)^1$   
 とする)

- 15:   If  $\forall \alpha_j < 120^\circ (0 \leq j \leq 2)$
  - 16:      $p_f$ へ移動
  - 17:   Else
  - 18:     If  $\alpha_i < 120^\circ$
  - 19:       ( $\alpha_j \geq 120^\circ$ となる点 $p_j$ を $p_v$ とする)
  - 20:        $p_v$ へ移動
  - 21:     Else
  - 22:       何もしない
- 

## 5.2. $n=4$ の場合の集合アルゴリズム

$n=4$  の場合の ASYNC, 局所多重検知, non-rigid, chirality で distinct gathering を解くアルゴリズムを Algorithm2 に示す.

アルゴリズムで用いる表記を以下の通りとする.

$SS_i$  : ロボット $r_i$ のスナップショット

$P(SS_i)$  :  $SS_i$ に含まれるロボットの位置の集合

$\mathcal{P}_n$  : 凸包の中の点の点を $p_r$ , 凸包の頂点を $p_1, p_2, p_3$ としたとき,  $\angle(p_{i-1}, p_r, p_{i+1}) \geq 90^\circ (0 \leq i \leq 2)^1$ となる $p_i$ の集合

図2に凸包の頂点を $p_0, p_1, p_2$ , 凸包の中の点が $p_r$ が存在し,  $\angle(p_0, p_r, p_1) = 85^\circ$ ,  $\angle(p_1, p_r, p_2) = 150^\circ$ ,  $\angle(p_2, p_r, p_0) = 125^\circ$ となる場合を示す. このとき,  $90^\circ$  以上となる角は $\angle(p_1, p_r, p_2)$ と $\angle(p_0, p_r, p_2)$ のため $\mathcal{P}_n = \{p_0, p_1\}$ となる.

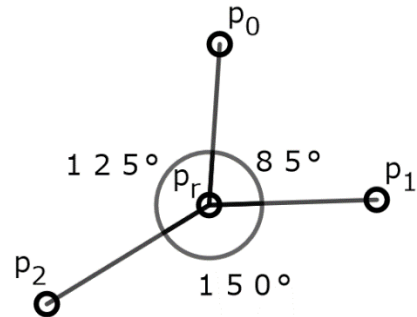


図 2 凸包の中に1台のロボットが存在する場合

---

<sup>1</sup> -, +は mod3 で計算する

$\mathcal{P}_s$  : 凸包の頂点の対角線の交点を $p_f$ とした時,  $p_f$ を中心として自身の時計回りの方向の点と作られた角が90度以上となる点の集合

図 3 に  $\angle(p_0, p_f, p_1) = \angle(p_2, p_f, p_3) = 110^\circ$  ,  
 $\angle(p_0, p_f, p_3) = \angle(p_1, p_f, p_2) = 70^\circ$  となる場合を示す. このとき,  $90^\circ$  以上となる角は $\angle(p_0, p_f, p_1) = \angle(p_2, p_f, p_3)$ のため $\mathcal{P}_s = \{p_0, p_2\}$ となる.

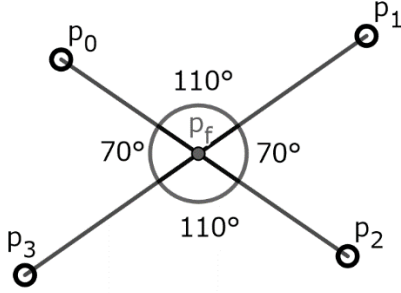


図 3 凸包の中にロボットが存在しない場合

---

Algorithm2. Distinct Gathering 4

---

仮定 :  $n=4$ , ASYNC, 局所多重検知, non-rigid, chirality

---

```

1: If  $|P(SS_i)|=2$ 
2:   If  $r_i$ が多重点
3:     停止
4:   Else
5:     他方に向かって進む
6: If  $|P(SS_i)|=3$ 
7:   If 3点が一直線上に存在する
8:     (3点を結ぶ線分の端点を $p_e$ , 内点を $p_m$ とする)
9:     If  $r_i = p_e$ 
10:       $p_m$ へ移動
11:     Else
12:      何もしない
13:   Else
14:     ( $P(SS_i)=\{p_0, p_1, p_2\}$ ,
        $\alpha_j = \angle(p_{j-1}, p_j, p_{j+1})^1$ とする)
```

```

15:   If  $\alpha_i < 120^\circ$ 
16:     ( $\alpha_j \geq 120^\circ$ となる点 $p_j$ を $p_v$ とする)
17:      $p_v$ へ移動
18:   Else
19:     何もしない
20: If  $|P(SS_i)|=4$ 
21:   If 4点が一直線上に存在する
22:     (4点を結ぶ線分の端点を $p_e$ , 内点を $p_m$ , 両端点 $p_e$ の中点を $p_c$ とする)
23:     If  $r_i = p_m$ かつ $r_i \neq p_m$ 
24:        $p_c$ へ移動
25:     Else
26:       何もしない
27:   Else If 3点が一直線上に存在する
28:     (3点を結ぶ線分の端点を $p_e$ , 内点を $p_m$ とする)
29:     If  $r_i \neq p_e$ かつ $r_i \neq p_m$ 
30:        $p_m$ へ移動
31:     Else
32:       何もしない
33:   Else
34:     凸包をつくる
35:   If 凸包の中に1台のロボットが存在する
36:     (凸包の中の点を $p_r$ ,
       凸包の周上の点を $p_0, p_1, p_2$ ,
        $\alpha_j = \angle(p_{j-1}, p_r, p_{j+1})$  ( $0 \leq j \leq 2$ )1,
        $\alpha_j \geq 90^\circ$  ( $0 \leq j \leq 2$ )となる $p_j$ の集合を $\mathcal{P}_n$ とする.)
37:     If  $r_i \in \mathcal{P}_n$ 
38:        $p_r$ へ移動
39:     Else
40:       何もしない
41:   Else
     ( $P(SS_i)=\{p_0, p_1, p_2, p_3\}$ ,  $\overline{p_0 p_2}$ と $\overline{p_1 p_3}$ 
       の交点を $p_f$ とする.
        $\alpha_j = \angle(p_{j-1}, p_r, p_{j+1})$  ( $0 \leq j \leq 3$ )1,
```

$\alpha_j \geq 90^\circ$  ( $0 \leq j \leq 3$ )となる  
 $p_j$ の集合を $\mathcal{P}_s$ とする. )

42: If  $r_i \in \mathcal{P}_s$   
 43:  $p_f \rightarrow$ 移動  
 44: Else  
 45: 何もしない

## 6. まとめ

本論文では  $n=3$  の場合の ASYNC, 局所多重検知, non-rigid で集合問題を解くアルゴリズムを提案した. また  $n=4$  の場合の ASYNC, 局所多重検知, non-rigid, chirality で distinct gathering を解くアルゴリズムを提案した. 今後の課題としては, Async モデルでの局所多重検知を用いて  $n \geq 5$  で集合を解くアルゴリズムの開発が挙げられる. またライト付きモデルでは Async モデルで未解決の部分の集合アルゴリズムの開発, 色数を減らした集合アルゴリズムの開発が課題として挙げられる.

## 参考文献

- [1] Paola Flocchini, Giuseppe Prencipe, Nicola Santoro. Distributed Computing by Oblivious Mobile Robots. SYNTHESIS LECTURES ON DISTRIBUTED COMPUTING THEORY, 2012
- [2] N. Agmon and D. Peleg, Fault-tolerant gathering algorithms for autonomous mobile robots, SIAM Journal on Computing, 36, 56{82, 2006.
- [3] T. Balabonski, A. Delga, L. Rieg, S. Tixeuil, X. Urbain, Synchronous gathering with-out multiplicity detection: a certified algorithm, Theory Compute. Syst., 63, 200-218, 2018.
- [4] Z. Bouzid, S. Das, and S. Tixeuil, Gathering of mobile robots tolerating multiple crash Faults. In Proceedings of 33rd ICDCS, 2013.
- [5] M. Cieliebak, P. Flocchini, G. Prencipe, and N. Santoro, Distributed computing by mobile robots: Gathering, SIAM Journal on Computing, 41(4): 829{879, 2012.
- [6] X. Defago, M. Gradinariu Potop-Butucaru, J. Clement, S. Messika, P. Raipin Parvedy: Self-stabilizing gathering of mobile robots under crash or Byzantine faults, Distributed Computing, pp. 1{29, doi:10.1007/s00446-019-00359-x, (2019).
- [7] 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.
- [8] Y. Dieudonne and F. Petit, Self-stabilizing gathering with strong multiplicity detection, Theoretical Computer Science, 428(13), 2012.
- [9] 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.
- [10] T Izumi, Y Katayama, N Inuzuka, and K Wada, Gathering Autonomous Mobile Robots with Dynamic Compasses: An Optimal Result, 21stDISC, 298{312, 2007.
- [11] T. Izumi, S. Souissi, Y. Katayama, N. Inuzuka, X. Defago, K. Wada, and M. Yamashita, The gathering problem for two oblivious robots with unreliable compasses, SIAM Journal on Computing, 41(1):26{46, 2012.
- [12] S. Kamei, A. Lamani, F. Ooshita, and S. Tixeuil, Asynchronous mobile robot gathering from symmetric configurations without global multiplicity detection, 18th SIR-ICCO, 150{161, 2011.
- [13] 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.

- [14] G. Prencipe, Impossibility of gathering by a set of autonomous mobile robots, *Theoretical Computer Science*, 384(2-3):222-231, 2007.
- [15] S. Souissi, X. Defago, 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.
- [16] S. Bhagat and K. Mukhopadhyaya. Optimum gathering of asynchronous robots. In *Conference on Algorithms and Discrete Applied Mathematics.*, pages 37-49, 2017.
- [17] P.Flocchini , N.Santoro , G.Viglietta ,M.Yamashita, Rendezvous with Constant Memory,*Theoretical Computer Science*,621, 57-72, 2016.
- [18] 亀田勇氣, 状態を持つ自律分散ロボット群に対する集合問題に関する研究, 法政大学大学院応用情報工学専攻 修士論文, 2020
- [19] T. Okumura, K. Wada, X. Defago, Optimal Rendezvous L-Algorithms for Asynchronous Mobile Robots with External-Lights, *OPODIS 2018, LIPICS*, vol. 125, pp. 24:1-24:16, (Dec. 2018).
- [20] S. Terai, K. Wada, Y. Katayama, Gathering Problems for Autonomous Mobile Robots with Lights, *arXiv:1811.12068*, 2018