

キーを持つロボットのソート問題における作業領域と実行時間

清水 貴弘^{*}; 山内 由紀子[†]

1 はじめに

分散アルゴリズム理論において、自律的に移動する匿名 (識別不能) なモバイルロボット群による**集合問題**や**パターン形成問題** [1] が研究されてきた。さらに、他のロボットが観測可能なライトを搭載した **ライト付きロボット** についての研究 [2] も行われてきた。ロボットは自身のライトの色を変更でき、ライトはロボットの状態や通信を表すと考えることができる。

本研究では、ロボットごとに固有の **キー** を持つロボットを、キーの順番に整列する **ソート問題** を提案し、ソート問題を解くための実行時間と実行中にロボットが移動する範囲である**作業領域**について調べる。本研究の目標は、作業領域と実行時間という観点から、モバイルロボットの分散協調問題の複雑性の新しい基準を提案することである。まず最初に、ロボットが自身のキーと他のすべてのロボットの位置とキーを観測できる場合を考え、作業領域が $n(n-1)$, $2n$, $3n$ の場合の完全同期モデルのロボットのためのソートアルゴリズムを提案し、その実行時間を評価した。さらに、他のロボットの位置は全て観測できるが、キーを観測できる範囲が制限されている場合のソート問題の可解性についても調べた。

2 準備

2.1 設定

整数値の**キー**をもつ n 台のロボットを考える。同じ値のキーを持つロボットはいないとする。ただし、1 から n である必要は無い。キーが i 番目に小さいロボットを r_i とする。ロボットの集合 R を $\{r_1, r_2, \dots, r_n\}$ とする。

各ロボットは直径 1 の円盤であり、重なろうとすると衝突する。また、ロボットは他のロボットの視界を妨げない。ロボット r_i の円盤の中心の座標を、 r_i が存在する位置とする。ロボットの視界は無限であり、他のロボットの位置とキーは観測できるとする。大域座標系におけるロボットの位置とキーの集合 $C = \{(x_i, y_i, k_i) \mid r_i \in R\}$ を**ロボットの配置**という。

各ロボット r_i は大域座標系 Z を知らず、自身の**局所座標系** Z_i で他のロボットの位置を観測する。 r_i の局所座標系は右手形の直交座標系とするが、その原点は r_i の現在の位置であり、 x 座標の方向と向きづけ、単位長は任意であるとする。

本研究では、各時刻 $t = 0, 1, 2, \dots$ に全てのロボットが同期して 観測-計算-移動サイクルを実行する **完全同期モデル**を想定する。観測フェーズでは各ロボットは自身の局所座標系で他のロボットの配置を観測する。計算フェーズでは、観測フェーズで得られた観測結果をもとに、各ロボットが自身の目的地を計算する。目的地を計算するアルゴリズムは全てのロボットで同一であり、 A とする。移動フェーズでは、各ロボットは計算した目的地に向けて移動する。ロボットは

^{*}九州大学工学部電気情報工学科

[†]九州大学大学院システム情報科学研究院

途中で止まらず必ず目的地に到達する．つまり，本研究が想定するロボットは，同一で，通信を行わず，無記憶である．

時刻 t の配置を C_t と表し，初期配置 C_0 から始まる配置の系列 C_0, C_1, \dots をアルゴリズム A の **実行** と呼ぶ．アルゴリズム A の **実行時間** を任意の初期配置から始まる実行のうち，最も長い実行の長さとする．アルゴリズム A の **作業領域** とは，任意の初期配置から始まる実行において，ロボットが移動する領域の大きさの最大値とする．

2.2 ソート問題

キーを持つ n 台のロボット $\{r_1, r_2, \dots, r_n\}$ が与えられた時，ロボット群がキーの順番に一直線上に並んだ配置を **ソート問題の終端配置** と呼ぶ．初期配置 C_0 から開始する実行 C_0, C_1, \dots が有限時間内にソート問題の終端配置に到達する時，この実行は **ソート問題を解く** と言う．与えられた初期配置 C_0 から始まるアルゴリズム A の任意の実行がソート問題を解く時， A は初期配置 C_0 からソート問題を解くと言う．

本研究では，ロボットが任意の順番で一直線上に並んだ初期配置のみを想定する．

3 ソート問題における作業領域と実行時間

以下では作業領域と実行時間の関係を調べるため，作業領域の異なる 4 つのアルゴリズムを観察する．

提案手法では，すべてのロボットが一直線上に並んだ初期配置のみを考える．この初期配置に対して，仮想的な座標系 Z^* を考える．初期配置において $p_1(0)$ から $p_n(0)$ に向かう直線を x 軸と考え， $p_1(0)$ から $p_2(0)$ へ向かう方向を x 軸の正の向きとする． x 座標は一番左から $1, 2, \dots, n$ とする．

ロボットが Z^* の x 座標に並行な一直線上に並び，にソートされているとき，**水平にソートされている** という．

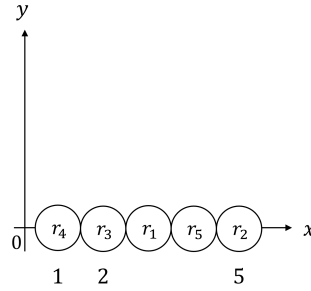


図 1: 大域座標系 Z^* における $x - y$ 軸

3.1 作業領域 $(n-1)n$ のソートアルゴリズム 1

このアルゴリズムでは段階的にソートを進めるため，前の段階の操作を行わないように以下のような配置の集合を考える．配置の集合は C_1, C_2, C_3, C_4 の順にソートが進んでいき， C_5 の状態になったときソートが完了となる．

- C_1 は以下の条件を満たす配置の集合である
 - r_1 が $(1, 0)$ にいない，もしくは r_n が $(n, 0)$ にいない，かつ

- 水平にソートされていない, かつ
- $r_i(2 \leq i \leq n-1)$ の y 座標が i もしくは 0
- \mathcal{C}_2 は以下の条件を満たす配置の集合である
 - r_1 が $(1,0)$ にいない, もしくは r_n が $(n,0)$ にいない, かつ
 - $r_i(2 \leq i \leq n-1)$ の y 座標が i
- \mathcal{C}_3 は以下の条件を満たす配置の集合である
 - r_1 は $(1,0)$, r_n は $(n,0)$ にいる, かつ
 - 水平にソートされていない, かつ
 - $r_i(2 \leq i \leq n-1)$ の y 座標が i
- \mathcal{C}_4 は以下の条件を満たす配置の集合である
 - r_1 は $(1,0)$, r_n は $(n,0)$ にいる, かつ
 - 水平にソートされている, かつ
 - $r_i(2 \leq i \leq n-1)$ の y 座標が i もしくは 0
- \mathcal{C}_5 は以下の条件を満たす配置の集合である
 - すべてのロボットが x 軸上にいる, かつ
 - 水平にソートされている

Algorithm 1 ロボット r_i におけるソートアルゴリズム *Sort_1*

```

1: //  $C$ : 観測結果
2:
3: if  $C \in \mathcal{C}_1$  then
4:   if  $2 \leq i \leq n-1$  then
5:      $y$  軸の正の向きに  $(i-1)$  移動する
6:   end if
7: end if
8:
9: if  $C \in \mathcal{C}_2$  then
10:  if  $i = 1$  or  $i = n$  then
11:     $i$  に移動する
12:  end if
13: end if
14:
15: if  $C \in \mathcal{C}_3$  then
16:  if  $2 \leq i \leq n-1$  then
17:     $x$  座標が  $i$  になるように  $x$  軸方向に移動する
18:  end if
19: end if
20:
21: if  $C \in \mathcal{C}_4$  then
22:  if  $2 \leq i \leq n-1$  then
23:     $y$  軸の負の向きに  $(i-1)$  移動する
24:  end if
25: end if

```

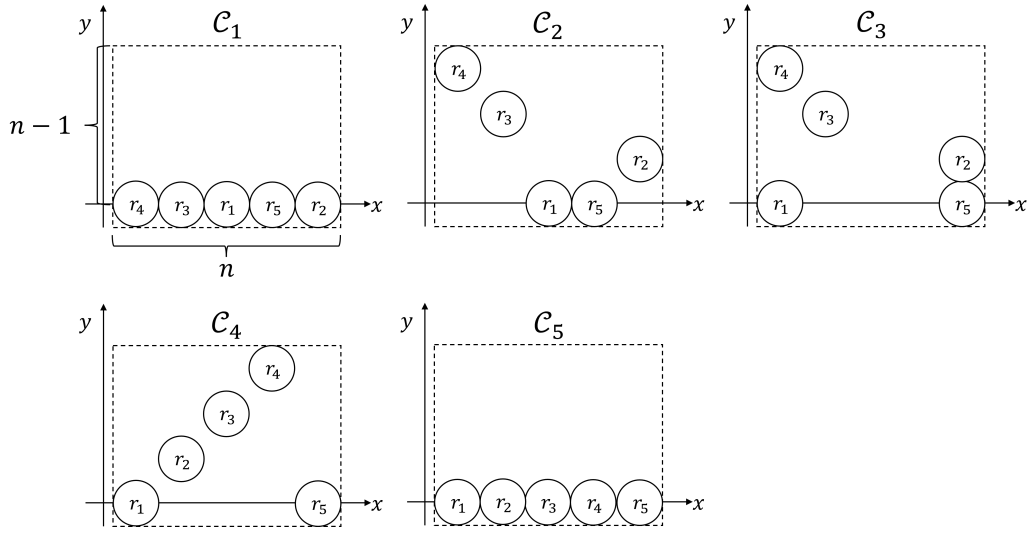


図 2: $n = 5$ のときのアルゴリズム 1 の動作例

定理 3.1 アルゴリズム 1 はすべてのロボットが一直線上に並んだ初期配置からソート問題を解く．作業領域は $(n-1) \times n$ であり，完全同期モデルでの実行時間は 4 ステップである．

3.2 作業領域 $2n$ のソートアルゴリズム 2

最小から順番にソートしていく．以下のようなロボットの配置の集合を考える． $\mathcal{C}_1, \mathcal{C}_2, \dots$ と昇順に並べられていく． \mathcal{C}_{n+1} となったときソートが完了となる．

- \mathcal{C}_1 : r_1 が $(1, 0)$ にいない
- $\mathcal{C}_i (i \geq 2)$: r_1 から r_{i-1} までソートされている

Algorithm 2 ロボット r_i におけるソートアルゴリズム $Sort_2$

```

1: //  $C$  : 観測結果
2:
3: if  $C \in \mathcal{C}_x (1 \leq x \leq n)$  then
4:
5:   if  $x = i$  then
6:     if 自分の  $x$  座標が  $i$  ではない and  $x$  軸上にいる then
7:        $y$  軸の正の向きに 1 移動し,  $x$  座標が  $i$  になるまで  $x$  軸の負の向きに移動する
8:
9:     else if 自分の  $x$  座標が  $i$  である and  $x$  軸上にいない then
10:       $y$  軸の負の向きに 1 移動する
11:    end if
12:
13:   else if  $x < i$  then
14:     if 自分の座標と  $(n, 0)$  の間に他のロボットがいない空間がある then
15:        $x$  軸正の向きに 1 移動する
16:     end if
17:   end if
18: end if

```

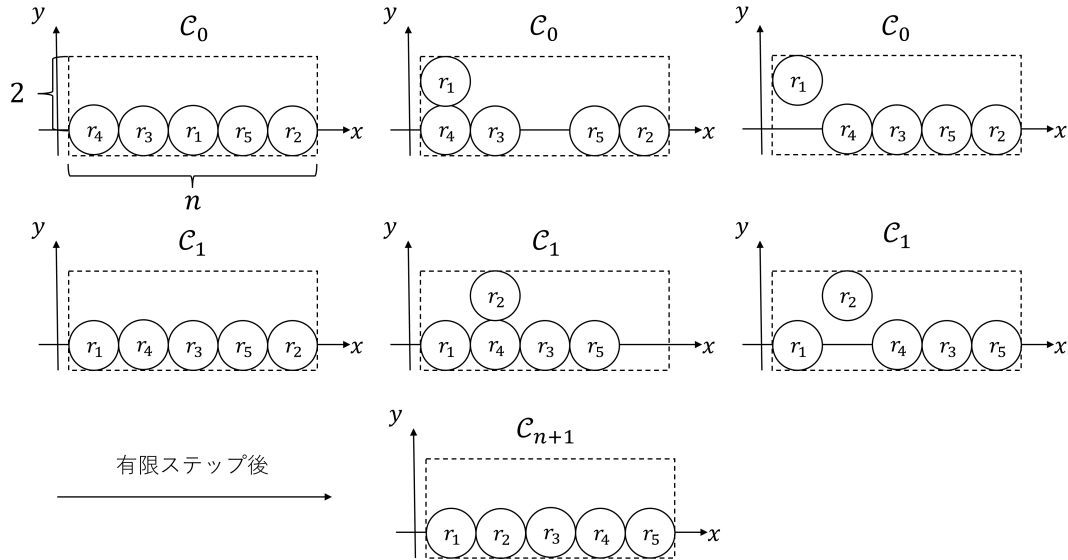


図 3: $n = 5$ のときのアルゴリズム 2 の動作例

定理 3.2 アルゴリズム 2 はすべてのロボットが一直線上に並んだ初期配置からソート問題を解く．作業領域は $2 \times n$ であり，完全同期モデルでの実行時間は $3(n-2)$ ステップである．

3.3 作業領域 $3n$ のソートアルゴリズム 3

以下のようなロボットの配置の集合を考える． C_1, C_2, \dots は，すべてのロボットが x 軸上にいたときに移動するロボットを決めるために設定する． C_{n+1} となったときソートが完了となる．

- \mathcal{C}_1 : r_1 が $(1, 0)$ にいない
- $\mathcal{C}_i (i \geq 2)$: r_1 から r_{i-1} までソートされている

Algorithm 3 ロボット r_i におけるソートアルゴリズム *Sort_3*

```

1: //  $\mathcal{C}$  : 観測結果
2:
3: if すべてのロボットが  $x$  軸上にある then
4:   if  $C \in \mathcal{C}_x (1 \leq x \leq n)$  and  $x = i$  then
5:      $y$  軸の正の向きに 1,  $x$  座標が  $i$  になるまで  $x$  軸方向に移動する
6:   end if
7: end if
8:
9: if ロボット  $r$  と  $r_i$  が同じ  $x$  座標にいる and  $r_i$  の  $y$  座標が 0 then
10:   $r$  がいる向きとは逆に  $y$  軸方向に 1,  $x$  座標が  $i$  になるまで  $x$  軸方向に移動する
11: end if
12:
13: if  $r_i$  の  $y$  座標が 0 ではない then
14:   $y$  座標が 0 になるまで  $y$  軸方向に移動する
15: end if

```

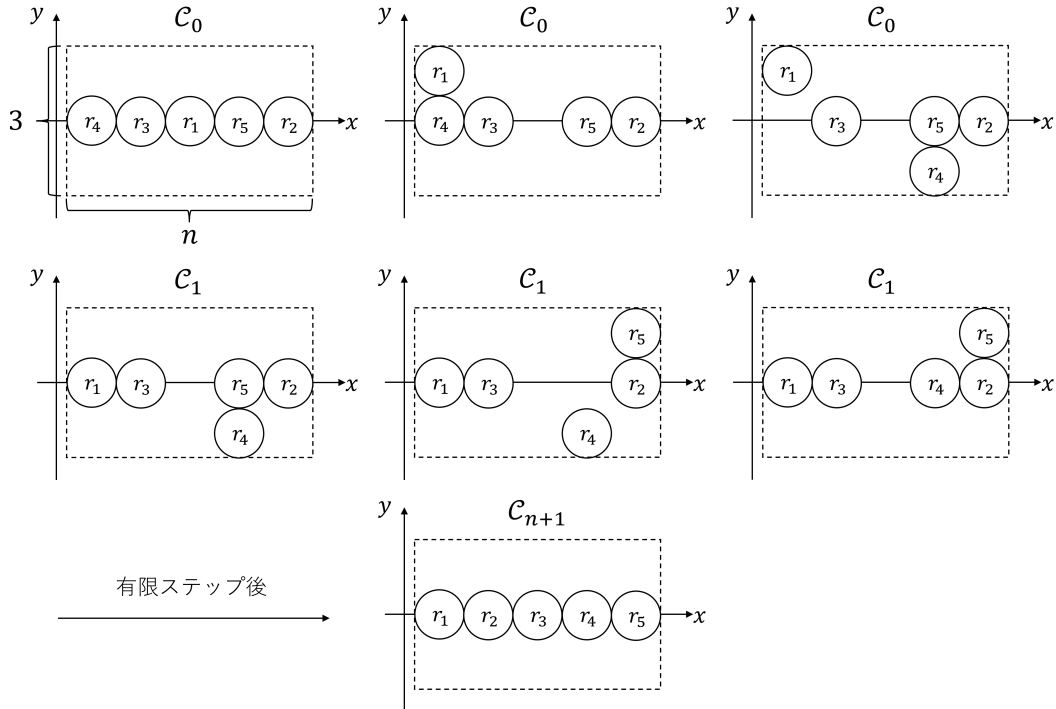


図 4: $n = 5$ のときのアルゴリズム 3 の動作例

定理 3.3 アルゴリズム 3 はすべてのロボットが一直線上に並んだ初期配置からソート問題を解く．作業領域は $3 \times n$ であり，完全同期モデルでの実行時間は $\frac{3}{2}n$ ステップである．

3.4 作業領域 $(\frac{n}{2} + 1)n$ のソートアルゴリズム 4

このアルゴリズムではまず中央値よりも大きいロボットを x 軸の正の向きに移動させてソートする．そして中央値以下のロボットを x 軸の負の向きに移動させる．これを実行するのは $\mathcal{C}_1, \mathcal{C}_2, \mathcal{C}_3$ の配置のときである．この段階が終わった状態の配置の集合を $Phase1_{fin}$ で表す．次に n が奇数だった場合には，キーが中央値となるロボットを中央に移動させる．これを行う配置が $\mathcal{C}_4, \mathcal{C}_5, \mathcal{C}_6, \mathcal{C}_7$ である． \mathcal{C}_8 では，中央値未満のロボットのソートが終わっていないためソートを実行する． \mathcal{C}_9 でソートが完了となる．

以下のような条件を満たす配置の集合， $Phase1_{fin}$ を考える．

- r_1 から $r_{\lceil \frac{n}{2} \rceil}$ の x 座標が 1 から $\lceil \frac{n}{2} \rceil$ である，かつ
- $r_{\lceil \frac{n}{2} \rceil + 1}$ から r_n の x 座標が $\lceil \frac{n}{2} \rceil + 1$ から n である，かつ
- $r_{\lceil \frac{n}{2} \rceil + 1}$ から r_n が水平にソートされている

以下のようなロボットの配置を考える．

- \mathcal{C}_1 は以下の条件を満たす配置の集合である
 - $r_{\lceil \frac{n}{2} \rceil + 1}$ から r_n の x 座標が $\lceil \frac{n}{2} \rceil + 1$ から n でない，または $r_{\lceil \frac{n}{2} \rceil + 1}$ から r_n が水平にソートされていない，かつ
 - r_1 から $r_{\lceil \frac{n}{2} \rceil}$ は x 軸上にいる，かつ
 - $r_i (\lceil \frac{n}{2} \rceil + 1 \leq i \leq n)$ の y 座標が $i - \lceil \frac{n}{2} \rceil$ もしくは 0，かつ
 - $r_{\lceil \frac{n}{2} \rceil + 1}$ から r_n の少なくとも一台は x 軸上にいる
- \mathcal{C}_2 は以下の条件を満たす配置の集合である
 - $r_{\lceil \frac{n}{2} \rceil + 1}$ から r_n の x 座標が $\lceil \frac{n}{2} \rceil + 1$ から n でない，または $r_{\lceil \frac{n}{2} \rceil + 1}$ から r_n が水平にソートされていない，かつ
 - r_1 から $r_{\lceil \frac{n}{2} \rceil}$ は x 軸上にいる，かつ
 - $r_i (\lceil \frac{n}{2} \rceil + 1 \leq i \leq n)$ の y 座標が $i - \lceil \frac{n}{2} \rceil$
- \mathcal{C}_3 は以下の条件を満たす配置の集合である
 - r_1 から $r_{\lceil \frac{n}{2} \rceil}$ の x 座標が 1 から $\lceil \frac{n}{2} \rceil$ ではない，かつ
 - $r_{\lceil \frac{n}{2} \rceil + 1}$ から r_n の x 座標が $\lceil \frac{n}{2} \rceil + 1$ から n である，かつ
 - $r_i (\lceil \frac{n}{2} \rceil + 1 \leq i \leq n)$ の y 座標が $i - \lceil \frac{n}{2} \rceil$
- \mathcal{C}_4 は以下の条件を満たす配置の集合である
 - $Phase1_{fin}$ ，かつ
 - r_1 から $r_{\lceil \frac{n}{2} \rceil}$ が x 軸上にいる，かつ
 - $r_i (\lceil \frac{n}{2} \rceil + 1 \leq i \leq n)$ の y 座標が $i - \lceil \frac{n}{2} \rceil$
- \mathcal{C}_5 は以下の条件を満たす配置の集合である

- $Phase1_{fin}$, かつ
 - すべてのロボットが x 軸上にいる, かつ
 - $r_{\lceil \frac{n}{2} \rceil}$ の x 座標が $\lceil \frac{n}{2} \rceil$ ではない
- \mathcal{C}_6 は以下の条件を満たす配置の集合である
 - $Phase1_{fin}$, かつ
 - $r_{\lceil \frac{n}{2} \rceil}$ を除くすべてのロボットが x 軸上にいる, かつ
 - $r_{\lceil \frac{n}{2} \rceil}$ が $(\lceil \frac{n}{2} \rceil, 1)$ にいる
- \mathcal{C}_7 は以下の条件を満たす配置の集合である
 - $Phase1_{fin}$, かつ
 - $r_{\lceil \frac{n}{2} \rceil}$ を除くすべてのロボットが x 軸上にいる, かつ
 - r_1 から $r_{\lfloor \frac{n}{2} \rfloor}$ の x 座標が 1 から $\lfloor \frac{n}{2} \rfloor$ である, かつ
 - $r_{\lceil \frac{n}{2} \rceil}$ が $(\lceil \frac{n}{2} \rceil, 1)$ にいる
- \mathcal{C}_8 は以下の条件を満たす配置の集合である
 - r_1 から $r_{\lfloor \frac{n}{2} \rfloor}$ の x 座標が 1 から $\lfloor \frac{n}{2} \rfloor$ である, かつ
 - $r_{\lfloor \frac{n}{2} \rfloor + 1}$ から r_n の x 座標が $\lfloor \frac{n}{2} \rfloor + 1$ から n である, かつ
 - r_1 から $r_{\lfloor \frac{n}{2} \rfloor}$ が水平にソートされていない, かつ
 - $r_{\lfloor \frac{n}{2} \rfloor + 1}$ から r_n が水平にソートされている, かつ
 - すべてのロボットが x 軸上にいる
- \mathcal{C}_9 は以下の条件を満たす配置の集合である
 - すべてのロボットが x 軸上にいる, かつ
 - 水平にソートされている

Algorithm 4 ロボット r_i におけるソートアルゴリズム *Sort_4*

```
1: //  $C$  : 観測結果
2:
3: if  $C \in C_1$  then
4:   if  $\lceil \frac{n}{2} \rceil + 1 \leq i \leq n$  then
5:      $y$  軸の正の向きに  $(i - \lceil \frac{n}{2} \rceil)$  移動する
6:   end if
7: end if
8:
9: if  $C \in C_2$  then
10:  if  $\lceil \frac{n}{2} \rceil + 1 \leq i \leq n$  then
11:     $x$  座標が  $i$  になるように  $x$  軸方向に移動する
12:  end if
13: end if
14:
15: if  $C \in C_3$  then
16:  if  $1 \leq i \leq \lceil \frac{n}{2} \rceil$  then
17:    1 もしくは, 衝突する手前まで  $x$  軸の負の向きに移動する
18:  end if
19: end if
20:
21: if  $C \in C_4$  then
22:  if  $\lceil \frac{n}{2} \rceil + 1 \leq i \leq n$  then
23:     $y$  軸の負の向きに  $(i - \lceil \frac{n}{2} \rceil)$  移動する
24:  end if
25: end if
26:
27: if  $C \in C_5$  then
28:  if  $i = \lfloor \frac{n}{2} \rfloor + 1$  then
29:     $y$  軸の正の向きに 1 移動し,  $x$  座標が  $i$  になるまで  $x$  軸の正の向きに移動する
30:  end if
31: end if
32:
33: if  $C \in C_6$  then
34:  if  $1 \leq i \leq \lfloor \frac{n}{2} \rfloor$  then
35:    1 もしくは, 衝突する手前まで  $x$  軸の負の向きに移動する
36:  end if
37: end if
38:
39: if  $C \in C_7$  then
40:  if  $i = \lfloor \frac{n}{2} \rfloor + 1$  then
41:     $y$  軸の負の向きに 1 進む
42:  end if
43: end if
44:
45: if  $C \in C_8$  then
46:  if  $1 \leq i \leq \lfloor \frac{n}{2} \rfloor + 1$  then
47:    1 から  $\lfloor \frac{n}{2} \rfloor + 1$  まででアルゴリズム 1 を実行する
48:  end if
49: end if
```

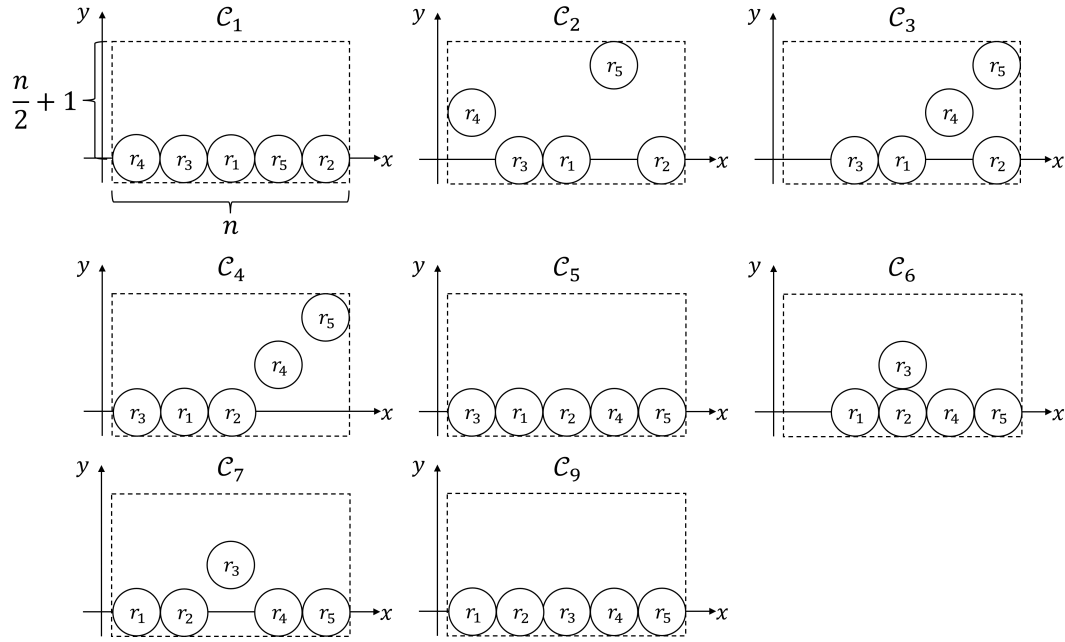


図 5: $n = 5$ のときのアルゴリズム 4 の動作例

定理 3.4 アルゴリズム 4 はすべてのロボットが一直線上に並んだ初期配置からソート問題を解く．作業領域は $(\frac{n}{2} + 1) \times n$ であり，完全同期モデルでの実行時間は 11 ステップである．

4 視界が有限の場合

キーを見られる範囲が有限の場合を考える (位置の視界は無限)．視界が $x (1 \leq x \leq n, x \in \mathbb{N})$ であるとは，一直線に並んだ初期配置において $(a, 0)$ にいるロボット r は， $(a - x, 0)$ から $(a + x, 0)$ までのロボットのキーしか見えないということ．

4.1 視界が $\lfloor \frac{n}{2} \rfloor$ の場合

中央のロボット r だけが全てのロボットのキーを見ることができる．

アルゴリズムの基本方針は，まず中央にキーが中央値のロボットを持って行く．すると，左右のグループではグループ内は全てのロボットが見えるので，ソートできる．このソートにはアルゴリズム 3.1 を使用する．また，どのロボットも全てのロボットの位置は見ることができるので，自分が中央なのかそうでないかわかる．それぞれのグループ内で以下の 5 つのパターンを考える．

1. r のキーより小さいロボットしかない
2. r のキーより大きいロボットしかない
3. r のキーより小さいロボットの数が多い
4. r のキーより大きいロボットの数が多い
5. r のキーより小さいロボットと大きいロボットの数が同じ

片方のグループが1のとき、もう片方は2になる。同じことが3と4にもいえるので、以下のようなロボットの配置が考えられる。

- C_1 : 1 と 2
- C_2 : 3 と 4
- C_3 : 5

アルゴリズムは以下のようになる。

1. 列の中央にあるロボットは中央値のロボットのところまで移動する。 n が偶数の場合は中央二つに中央に最も近い2つの値が来るように移動する。
2. それぞれのグループで以下のように移動する。
 - C_1 の場合, 1 のグループは r が最大になるようにソートする。2 のグループは r が最小になるようにソートする。
 - C_2 の場合, 3/4 の場合, r に近い方が大きく/小さくなるようにソートする。3/4 のグループの中の最大/最小のロボットは r に向かって左に1進み, r のいる方に x 軸方向に2進む。そして, x 軸上に移動する。
 - C_3 の場合, r に近い方が大きくなるようにソートする。 r の両隣の内、大きいロボットが移動する。 r に向かって左に1進み, もう片方のグループの内, r より小さいロボットの中で最大のロボットのいるところまで x 軸方向に移動する。

定理 4.1 アルゴリズム 4.1 はすべてのロボットが一直線上に並んだ初期配置からソート問題を解く。作業領域は $\frac{n}{2} \times n$ であり, 完全同期モデルでの実行時間は 14 ステップである。

4.2 視界が1の場合

視界が1の場合, 初期配置ではロボット r は両隣のロボットしか見えない。

定理 4.2 視界が1のとき, 隣と位置を交換する方法によってソート問題を解くことはできない。

証明. この場合, 自分から見える範囲では以下の4パターンしかない。

1. 昇順
2. 降順
3. 谷 (見える範囲で自分が一番小さい)
4. 山 (見える範囲で自分が一番大きい)

ソートが完了していた場合はロボットが停止する必要があるので, 1と2のパターンはロボットが移動しない。よって, 3と4のみを考えれば良い。

1. 3もしくは4のときのみ動く場合

3の場合を考える。両端に最小と二番目に小さいロボットがいるとする。このときこのロボットは動かないので, 3と4の両方の場合で動く必要がある。

2. 3 と 4 のときの両方で動く場合

同じキーはないので、3 もしくは 4 のときには、左右のキーの大小がわかる。キーが "4, 1, 2, 3" という並びを考える。このとき 1 が谷になっている。左右のキーの内、小さい方と交換すると、"4, 2, 1, 3" となる。ここでも 1 が谷になっているので同じ操作を行うと "4, 1, 2, 3" という元の並びに戻ってしまう。よって、谷の時左右のキーの内、小さい方と交換する方法は使えない。4 の山の場合も同様に左右のキーの内、大きい方と交換する方法が使えない。

逆に谷の時は左右のキーで大きい方と、山の場合は左右のキーで小さい方と交換するとする。"2, 1, 4, 3" を並び替えると "2, 4, 1, 3" となる。もう一度同じ操作を行うと元の並びに戻る。

よって視界が 1 の場合、隣と交換という方法によってソート問題を解くことはできない。

□

5 まとめ

本稿では、作業領域を変えて実行時間を調べた結果、トレードオフの関係が確認できた。しかし、今回提案したアルゴリズムがそれぞれの作業領域に対して最適なものはわかっていない。そのため今後はそれぞれの作業領域における実行時間の下界を調べたい。

また、キーを観測できる範囲が制限されている場合のソート問題の可解性を調べ、視界が 1 のとき隣との交換によってソートができないことが分かった。今後は、視界が制限されている場合のソート問題の実行の可否をより詳しく調べたい。

References

- [1] Ichiro Suzuki and Masafumi Yamashita, "Distributed Anonymous Mobile Robots: Formation of Geometric Patterns," SIAM Journal on Computing, Vol. 28, Issue 4, pp. 1347–1363, 1999.
- [2] Shantanu Das, Paola Flocchini, Giuseppe Prencipe, Nicola Santoro, and Masafumi Yamashita, "Autonomous mobile robots with lights," Theoretical Computer Science, Vol. 609, Part 1, pp. 171–184, 2016.