共通座標系を持たない5台の非同期式ファットロボットによる集合問題について

平野　拓弥[1]　片山 喜章[1]　和田　幸一[2]

概要：本稿では，連続平面上において大きさを持つ 5 台の自律分散ロボットによる集合問題を扱う．ロボットは共通の座標系を持たず，匿名で，通信を行わない．透明で視界距離に関しての制限を持たず，非同期に動作する．ロボットは他のロボットの視界を遮らないが，移動時の障害物となりうるため，衝突を防ぐアルゴリズムの設計が重要となる．同モデルにおける n 台の集合問題に対するアルゴリズムはすでに提案されているが [8]，このアルゴリズムには問題点が存在する．そこで，我々は台数を 5 台に限定することで文献 [8] とは異なる手法で問題を解決した．

1. はじめに

近年，自律分散ロボット群の研究が盛んに行われている．自律分散ロボットとは自律分散システムの一つで，低機能なロボットを複数台用いて，協調して動作させることでロボット群全体で一つの目的を達成させるシステムである．ロボット能力を低く抑えることで，コストの低く抑え，システムの拡張を容易にできるといった利点がある．また，それぞれのロボットが自律的に動作するため，人による制御が難しい場所でも安定して動作することができる．研究の焦点はロボットの持つ情報や能力と問題の可解性との関係であり，より低機能なロボットで問題を解くことが目標とされている．

ロボットは他のロボットの位置を観測し (Look)，観測した情報を用いてアルゴリズムに従って移動先を計算し (Compute)，計算した移動先に移動する (Move)，というサイクルを繰り返して問題を解決する．全てのロボットは同じアルゴリズムで動作する．よく扱われる問題として，一点集合問題が挙げられる．一点集合問題とは，あらかじめ決められていない一点に全てのロボットを集合させる問題である．

集合問題の中でも，2 台のロボットの集合を考えた問題を特にランデブーと呼ぶ．ランデブーは特別な能力がなければ集合を達成できないことは広く知られており，コンパスやメモリを有したロボットを用いて研究が行われている [2][3]．また，3 台以上のロボットによる集合問題をギャザリングと呼ぶ．こちらは，重複検知能力を持たせることで解決可能であると証明されている [4]．

純粋に理論的な研究は，ロボットを平面上を動く点として表現してきた．一方で，現実のロボットシステムへの適用を考える場合，より現実的なモデルの導入が必要である．そこでロボットの持つコンパスやセンサの誤差を考慮したモデル [2] や，ロボットの視野に制限を設けたモデル [5] などの研究も行われている．さらに，現実のロボットは大きさを持つため，ロボットが他のロボットの観測および移動について障害物となりうる．そこで，ロボットを点ではなく円盤で表したモデルであるファットロボット (fat robot) も提案されており，ファットロボットに関しても研究が進められている．ファットロボットは大きさを持ち，お互いに重なりあうことができないため，2 台以上のロボットが同一点を共有することは不可能である．よって，集合の定義についても再考慮する必要がある．

文献 [6] は最初にファットロボットの集合問題を扱った論文である．ロボットが 3 台または 4 台の場合についてのアルゴリズムを提案している．また，文献 [7] はキラリティを持つ 5 台以上のファットロボットに関しての集合問題を取り扱っている．どちらの論文も，ロボットは他のロボットの視界を遮る (不透明)，という条件の元でアルゴリズムを設計しており，集合の条件は「全てのロボットが連結であり，互いに見ることができる」というものである．文献 [10] では，離散平面上でのファットロボットの集合問題に関しての研究が行われている．

我々はキラリティを持たず，ロボットが他のロボットの

[1] 名古屋工業大学大学院工学研究科情報工学専攻
Nagoya Institute of Technology, Graduate School of Computer Science and Engineering
[2] 法政大学理工学部応用情報工学科
Hosei University, Faculty of Science and Engineering Department of Applied Informatics

視界を遮らない，つまりロボットを透明と仮定したモデルについてのアルゴリズムを提案する．我々と同様のモデルにおいて，任意の台数のロボットの集合問題に関する研究が文献 [8] でなされているが，この論文中で提案されているアルゴリズムは二つの問題点がある．一つ目は，最初に集合の中心となる部分に 1 台のロボットが移動する際に，他のロボットに移動を妨げられる場合を考慮していない点．二つ目は，最後の数台が集合する際にロボットの配置が対称となる可能性があり，移動するロボットが一意に決定できず，想定通りの動作を実行できない点である．

我々は，ロボットの台数を 5 台に限定することで，文献 [8] とは異なる手法を用いて集合を達成した．

2. モデルと問題定義

2.1 ロボットのモデルと仮定

5 台のファットロボットの集合を $R = \{r_1, r_2, ..., r_5\}$ で表す．ロボットは半径 1 の単位円であり，連続二次元平面上を自由に移動する．このとき，ロボットには大きさがあるため，移動時に他のロボットと衝突しうる．また，ロボットは任意の直線もしくは円周に沿って移動ができるものとする．ロボットの移動は目的地まで到達する前に任意の位置で停止しうるが，少なくとも最低移動距離 Δ 以上は移動できるとする．なお，ロボットの位置はロボットを表現する単位円の中心点と考える．したがって，例えば互いに接する 2 台のロボット間の距離は 2 である．

ロボットは匿名で，外観や識別子等によって他のロボットと区別することはできない．また，他のロボットと直接通信を行うこともできない．自身の持つカメラやセンサ等を用いて他のロボットの位置を観測することでのみ，他のロボットの情報を得ることができる．

ロボットは共通の座標系 (軸，原点，キラリティ等) に関する一切の知識を持たないが，ロボットの半径が同一なため，単位長についての合意は持っているとする．ロボットは観測した他のロボットの位置情報を自身のローカル座標系に当てはめ，その情報をもとに移動先を計算し，移動する．ロボットは他のロボットのローカル座標系について知ることはできない．

ロボットは透明であり，他のロボットの視界を妨げない．また，視野範囲や視界距離は無限であり，他のロボット全ての位置を正確に観測できる．ロボットは過去の動作を記憶できず，常に最新の観測結果のみに基づいて行動する．また，全てのロボットは同一のアルゴリズムで動作する．

ロボットは Look，Compute，Move の 3 つのフェイズから成るサイクルを繰り返して問題を解決する．Look フェイズでは，他のロボットの位置を自身の持つカメラやセンサ等で観測する．Compute フェイズでは，観測した情報をもとにアルゴリズムを実行し，自分の移動先を決定する．Move フェイズでは，Compute フェイズで決定した移動先に向かって移動する．サイクルを実行するタイミングについては 3 つのモデルが提案されており，それぞれ完全同期 (FSYNC)，半同期 (SSYNC)，非同期 (ASYNC) と呼ぶ．

図 1　dns=7 のときの図

FSYNC モデルでは，全てのロボットが完全に同じタイミングでサイクルの各フェイズを実行する．SSYNC モデルでは，FSYNC と同様に各ロボットは同じタイミングでサイクルを実行するが，その際に動作しないロボットが存在する．つまり，ロボットの部分集合が完全に同じタイミングで動作する．ASYNC モデルでは，ロボットのサイクルの実行に関して一切の仮定を置かない．よって，例えばあるロボットが Compute しているタイミングで別のロボットが Move をしていると，古い位置情報に従って移動先を決定し移動する可能性がある．本研究では，ASYNC モデルを仮定する．

また，初期状況として，接し合っている (=距離が 2 である) ロボットは存在せず，全てのロボットは静止しているものとする．

2.2 問題定義

ロボット間の距離が 2 である組の数を密度といい，dns で表す．密度が最大である状態で集合することを，本研究での達成目標とする．ロボットの台数が 5 台の場合について，密度が最大となる状態を**図 1** に示す．図からわかる通り，ロボットの台数 n が 5 のとき，密度の最大値は $dns = 7$ である．

定義 2.1. 集合問題：5 台のロボットにおいて，dns=7 の状態で全てのロボットが静止したとき，ロボットは集合したという．

2.3 諸定義

ロボット r 上で実行されるアルゴリズムで用いる変数や関数などを以下に定義する．

- $GatheringSet$ ： 互いに接し合っているロボットの集合．
- L_1 : $dns = 1$ のとき，$GatheringSet$ に含まれる 2 台のロボットの接点を通る共通接線 (**図 2**)．
- L_2 : $dns = 1$ のとき，$GatheringSet$ に含まれる 2 台のロボットの中心を通る直線 (図 2)．
- L_3 : $dns = 2$ であり，$GatheringSet$ に含まれる 3 台が一直線上に並んでいるとき，その 3 台の中心を通る

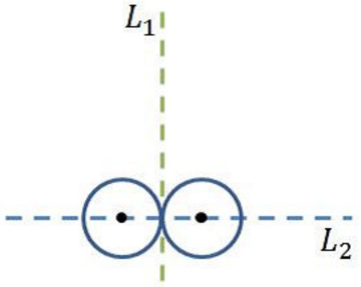

図 2　L_1 と L_2

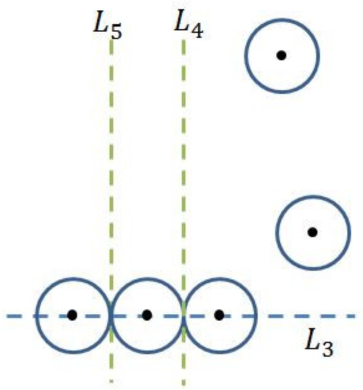

図 3　L_3 と L_4 と L_5

直線 (**図 3**).

- L_4, L_5：$dns = 2$ であり，$GatheringSet$ に含まれる 3 台が一直線上に並んでいるとき，接し合っているロボットの組それぞれの共通接線のうち，$GatheringSet$ に含まれない 2 台のうち，L_3 により近い方ロボットに対して，近い方の直線を L_4，遠い方の直線を L_5 という (図 3).
- $dist(r, L_i)$：ロボット r と直線 L_i との距離.
- $NotOnCircle(R)$：5 台全てのロボットが同一円周上に存在する場合には false，それ以外の場合 true となる boolean 関数.
- $Pent(R)$：同一円周上に 5 台のロボットが存在しているとき，その 5 台による凸五角形.
- $Edge(r_i)[4]$：$Pent(R)$ の最短辺を成すロボット r_i の，最短辺の逆方向周りの 4 本の辺の長さを自分に近い順に格納する配列.
  図 4 の例において，$Edge(r_0)$ は
  $Edge(r_0)[0] = edge_0, Edge(r_0)[1] = edge_1,$
  $Edge(r_0)[2] = edge_2, Edge(r_0)[3] = edge_3$
  である．同様に，$Edge(r_4)$ は
  $Edge(r_4)[0] = edge_3, Edge(r_4)[1] = edge_2,$
  $Edge(r_4)[2] = edge_1, Edge(r_4)[3] = edge_0$
  である．ただし，$edge_i$ はそれぞれの辺の長さであり，最短辺を成さないロボット r_i の $Edge(r_i)$ は NULL とする.

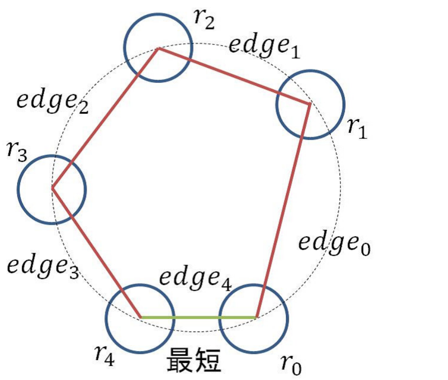

図 4　同一円周上に 5 台のロボットが存在する場合の例

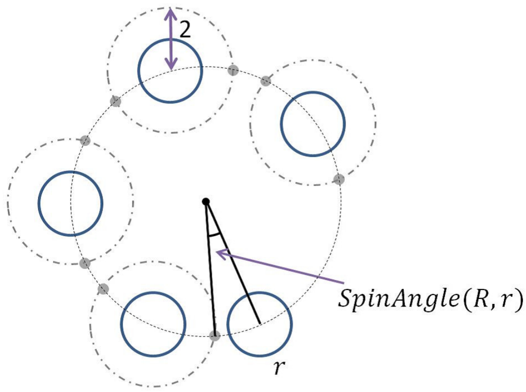

図 5　$SpinAngle(R, r)$ の例

- SS：$Pent(R)$ の最短辺を成すロボットの集合.
- $SpinAngle(R, r)$：5 台のロボットが同一円周上に存在するとき，その円周上において，$r_i \in R - \{r\}$ からの距離が 2 である点の中で r に最も近い点と，r が成す角度 (**図 5**).
- $Spin(angle)$：同一円周上に 5 台のロボットが存在するとき，その円周上に沿って角度 angle だけ最も近いロボットの方向へ移動する命令.
- $Gather()$：L_2 上かつ，ロボット $r' \in GathringSet$ と接する点のうち，最も近い点へ直線移動する命令.
- $Slide(length, L_i, dir)$：距離 $length$ だけ，直線 L_i の垂直方向に，dir 向き (far : L_i から遠ざかる，もしくは $near$: L_i に近づく) に直線移動する命令.

3. アルゴリズム

3.1 基本戦略

アルゴリズムの基本的な戦略は以下の通りである.

(1) 全てのロボットを同一円周上へ移動させる
(2) 同一円周上にある 5 台の中から 1 台を移動させ，2 台を接触させる
(3) $GatheringSet$ に含まれていない 3 台のうちの 1 台を，接触している 2 台のロボットと一直線になるように接触させる

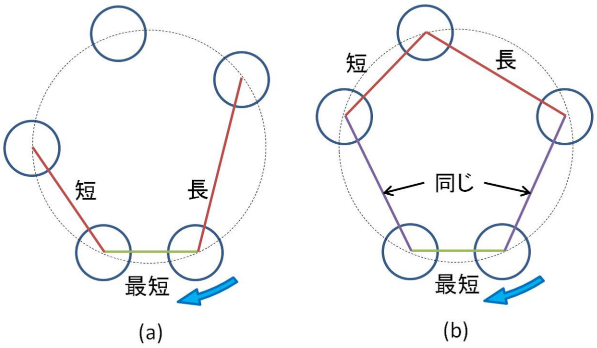

図 6　dns=0 の場合に移動するロボット

(4) 残り 2 台のロボットを順番に集合させる
全てのロボットを最初に同一円周上へ移動させることで，初期状況に関わらずその後のアルゴリズムの設計や正当性の証明を考えることができる．同一円周上へ全てのロボットが移動した後は，移動時のロボット同士の衝突を防ぐため，1 台のロボットのみが移動するようにアルゴリズムを設計する．1 台のロボットのみを移動させるためには，対称性の発見とその解消が重要となる．

3.1.1　同一円周上への移動

同一円周上への移動に関しては，円形成アルゴリズムを利用することで容易に達成することができる．文献 [9] において，本研究と同じモデルで円形成問題を解決するアルゴリズムが提案されているため，そちらを参照する．$dns = 0$ かつ 5 台のロボットが同一円周上に存在しない場合に限り，円形成を行う．

3.1.2　2 台の接触

$dns = 0$ かつ，同一円周上に 5 台のロボットが存在しているとき，1 台がその円周に沿って移動し，2 台が接触している状況を作り出す．移動するのは，同一円周上にある 5 台で凸五角形を計算し，その最短辺を構成するロボットのうちのいずれかとする．このとき，ロボットが移動することで，移動する条件に合致するロボットが変化しないように注意する必要がある．ASYNC モデルではロボットが移動中に他のロボットによって観測されうるため，もしロボットの移動中に，そのロボットの移動によって条件に合うロボットが変化した場合，複数のロボットが移動し，結果的に不都合な状況が生じる可能性がある．それを防ぐため，最短辺を構成するロボットのうち，最短辺と逆側の辺の長さが最も大きいロボットが移動することとする (**図 6**-a)．もしこの値が等しい場合，最短辺の逆側の辺を更に 2 本目，3 本目と比較していき，辞書式順で最大となるロボットが移動する (図 6-b)．

このとき，ロボットの配置が対称 (線対称・回転対称) の場合，移動するべき 1 台のロボットを一意に決定することができない可能性がある．ただし，今回は台数を 5 台に限定したため，回転対称となるのはロボットの配置が正五角形になっている場合のみであり，この場合は集合の達成は不可能とする．また，線対称については対称軸上に 1 台のロボットが存在し，残りの 4 台のロボットが対称軸を挟んで 2 台ずつ，対称の位置に存在する場合に限られる．よって，対称軸上のロボットが一意に決定され，そのロボットが移動することで，対称性を崩すことが可能である．対称軸上のロボットは，任意の方向に円周に沿って微小距離移動する (**図 7**)．ただし，微小距離を移動する途中で再び対称となる場合は，対称となるポイントの直前で停止することで対称となるのを防ぐ．非対称な形状であれば動くべき 1 台を一意に決定することが可能となる．

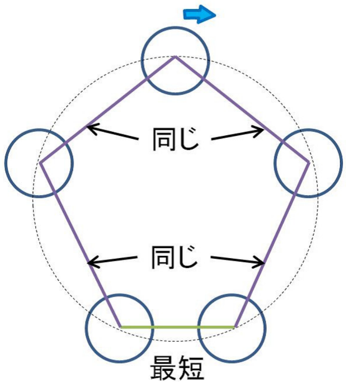

図 7　dns=0 かつ対称の場合に移動するロボット

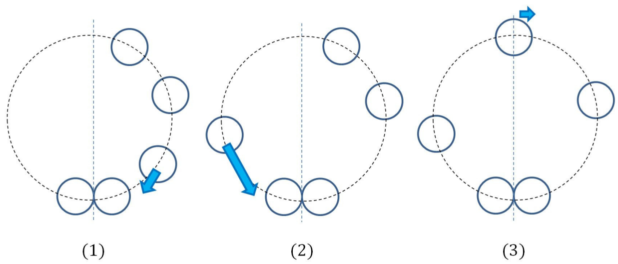

図 8　dns=1 の場合の 3 つのパターン

3.1.3　3 台目の移動

一組のロボットが接触した後，その 2 台のロボットによって定義される直線 (L_1, L_2) を基準として残りの 3 台を順番に集合させる．$GatheringSet$ に含まれていない 3 台のロボットの中から 1 台のロボットが移動するが，このとき，L_1 を基準として，3 台のロボットの配置については 3 つのパターンが考えられる (**図 8**)．

(1) 3 台のロボットが L_1 を基準にして同じ側に存在する
(2) L_1 によって 2 台のロボットと 1 台のロボットに分けられる
(3) L_1 上に 1 台のロボットが存在する

この時点では全てのロボットは同一円周上に存在するため，その円の直径である L_1 上に複数のロボットが存在することはない．また，パターン 3 については，L_1 上に位置するロボットが L_1 に対して垂直方向に，任意の向きに移動することで，パターン 1 かパターン 2 に推移することができる．

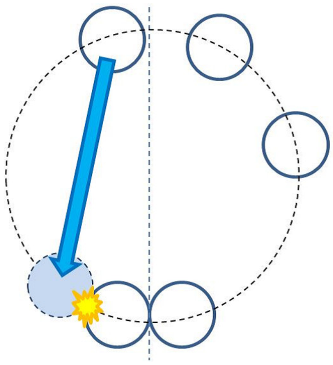

図 9　衝突が発生する例

パターン 1 の場合については，3 台のうち，L_2 に最も近いロボットが移動し集合する．移動先は L_2 上の，$GatheringSet$ に含まれるロボットに接することができる点のうち，自分に近い方の点である．このとき，全てのロボットは同一円周上にいるため他のロボットに移動を妨げられることはない．(図 8-1)

パターン 2 の場合については，L_1 を基準にして 1 台のみが存在する側のロボットが移動する．この場合も移動先はパターン 1 と同様である．ただし，ロボットと L_1 との距離が 3 未満の場合，他のロボットと衝突する可能性がある (**図 9**)．よって，その場合は L_1 との距離が 3 となる位置へ移動してから集合することで解決する．

このように移動することで，3 台が一直線上 (L_3 上) に並んだ状態で，残りの 2 台は L_3 からの距離が異なる状況を作り出すことができる．よって，これを利用して残りの 2 台を順序付けすることができる．

3.1.4　4 台目と 5 台目の集合

残り 2 台を集めるとき，L_3 から遠いロボットから移動する．もし L_3 からの距離が 4 未満の場合，L_3 から 2 離れる．その後，L_5 上へ移動する．L_3 に近い方のロボットは，遠い方のロボットが L_5 上に位置しないとき静止しつづける (**図 10**-a)．遠い方のロボットが L_5 上に位置する場合は，L_4 上へ移動する (図 10-b)．このとき，他のロボットに移動を妨げられる場合 (L_3 との距離が 2 未満の場合) は，L_3 から 2 離れることで解決する．このとき，遠い方のロボットと L_3 との距離は必ず 4 以上であるので，近い方のロボットと遠い方のロボットの L_3 に対しての位置関係が変わることはない．

L_3 からの距離が遠いロボットが L_5 上に，近い方のロボットが L_4 上に乗っている場合，L_3 に近いロボットが先に L_4 に沿って集合する．最後の 1 台は，自分以外の全てのロボットが集合している状態でのみ，集合するために移動する．

以上の基本戦略を実現したアルゴリズムを**図 11**，**図 12** に示す．

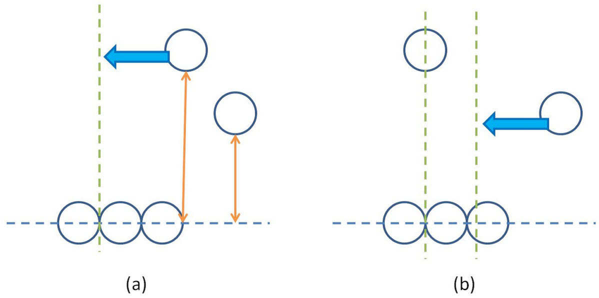

図 10　dns=2 の場合に移動するロボット

3.2　正当性の証明

アルゴリズムを実行することで，有限時間内に 5 台のロボットが dns=7 の状態で静止することを示す．

補題 3.1. dns=0 かつ $NotOnCircle(R) = true$ のとき，有限時間内に dns=0 かつ $NotOnCircle(R) = false$ となる

証明 1. 仮定より，初期状況では接しているロボットは存在しないため，dns=0 である．アルゴリズムより，dns=0 かつ $NotOnCircle(R) = true$ のとき，ロボットは文献 [9] に示された円形成アルゴリズムを実行する．円形成アルゴリズムが有限時間内に完了することは，文献 [9] による．

よって，dns=0 かつ $NotOnCircle(R) = true$ のとき，全てのロボットは有限時間内に同一円周上へ移動する．つまり，有限時間内に dns=0 かつ $NotOnCircle(R) = false$ となる． □

補題 3.2. dns=0 かつ $NotOnCircle(R) = false$ かつ $Pent(R)$ が正五角形でないとき，有限時間内に dns=1 になる

証明 2. $NotOnCircle(R) = false$ であるとき，5 台のロボットは凸五角形を成している ($Pent(R)$)．$Pent(R)$ が非対称の場合，各ロボットが自分から見て時計回り・反時計回りに $Pent(R)$ の辺を並べたとき，その系列は必ず異なる．ロボットは他の全てのロボットの位置情報を持つため，他ロボットから見た時計回り・反時計回りの辺の系列も計算することが可能で，各系列はそれぞれ異なっている．

Require: ロボットの集合 R, ロボット r
Ensure: Boolean 変数 (true,false)
　$Pent(R)$ の最短辺を成すロボットの集合 SS を調べる
　if r が成す辺が最短辺 **then**
　　$j \Leftarrow 0$
　　while $r \in SS$ **do**
　　　$\forall r_i \in SS$ について，$Edge(r_i)[j]$ の値を比較
　　　if $Edge(r)[j]$ が唯一の最大値 **then**
　　　　return(true)
　　　end if
　　　$Edge(r_i)[j]$ が最大値でない r_i を SS から取り除く
　　　$j++$
　　end while
　end if
　return(false)

図 11　CheckMove(R,r)

よって，自分を含めた全てのロボットが持つ系列を比較可能であり，それを用いてロボットに順番づけをし，全ロボットが共通の 1 台を選択することができる．アルゴリズムより，$Pent(R)$ の最短辺を成すロボットは，最短辺を成す全てのロボットの $Edge(r_i)$ の値を計算・比較することで，移動するべき条件を満たす 1 台のロボットを一意に決定し，そのロボットのみが最短辺の向きに移動する．よって，最短辺が複数あった場合でも，$Pent(R)$ が非対称ならば移動するべき唯一のロボットが決定可能であり，さらに移動の最中にも移動する条件を満たすロボットは変化しない．また，移動は 5 台のロボットが乗っている円周に沿って行われ，移動距離は有限なので，有限時間内に 2 台のロボットの距離が 2 となる．よって，5 台のロボットが成す $Pent(R)$ が非対称の場合，1 台のロボットのみが移動し，有限時間内に dns=1 になる．

また，$Pent(R)$ が線対称でありかつ正五角形でない場合，その形状は 1 台のロボットのみが対称軸上に存在する線対称に限られる．よって，対称軸上のロボットが円周に沿って任意の方向へ微小距離移動することで，形状を非対称にすることができる．

以上から，dns=0 かつ $NotOnCircle(R) = false$ かつ $Pent(R)$ が正五角形でないとき，有限時間内に dns=1 になる． □

補題 3.3. dns=1 のとき，有限時間内に dns=2 になる

証明 3. アルゴリズムにより，dns=1 になった瞬間，全てのロボットは同一円周上に存在する．よって，このときのロボットの配置は以下の 3 パターンに分けられる．

(1) 3 台のロボットが L_1 を基準にして同じ側に存在する

(2) L_1 によって 2 台のロボットと 1 台のロボットに分けられる

(3) L_1 上に 1 台のロボットが存在する

どのパターンについても，移動するべきロボットを一意に決定可能なのは明らかである．パターン 1 については，L_2 まで最も近いロボットについて目的地まで障害物が存在せず，直接移動可能なのは明らかであり，また移動の最中でも移動する条件を満たすロボットは変化しない．パターン 2 については，目的地まで直接移動できない可能性はあるが，L_1 から離れる向きに移動することでそれを解決できる．その際にも移動する条件を満たすロボットは変化せず，また，目的地まで移動できるのは明らかである．パターン 3 については，L_1 上のロボットが微小距離移動することでパターン 1 かパターン 2 に移行する．どの場合においても移動距離は有限であるため，移動が有限時間内に終わることは明らかである．

以上から，dns=1 のとき，有限時間内に dns=2 になる． □

補題 3.4. dns=2 のとき，有限時間内に dns=4 となる

証明 4. dns=1 のとき L_1 上にロボットがいない場合，dns=2 になった時点で残り 2 台が L_3 からの距離が異なるのは明らかである．また，L_1 上にロボットが存在する場

Require: ロボットの集合 R, ロボット r
Ensure: ロボットの移動先 T
  if $r \in GatheringSet$ **then**
    exit
  else if R が正五角形 **then**
    exit
  else if $dns = 0$ **then**
    if $NotOnCircle(R)$ **then**
      $CircleFormation(R, r)$
    else if R が対称 **then**
      if $dist(r, L_0) = 0$ **then**
        $Spin(\delta)$
      end if
    else if $CheckMove(R, r)$ **then**
      $Spin(SpinAngle(R, r))$
    end if
  else if $dns = 1$ **then**
    if パターン 3 **then**
      if $(dist(r, L_1) = 0)$ **then**
        $Slide(\delta, L_1, far)$
      end if
    else if パターン 1 **then**
      if $GatheringSet$ に含まれていない 3 台に関して，L_1 を基準として自分と同じ側にロボットが存在しない **then**
        if $dist(r, L_1) < 3$ **then**
          $Slide(3 - dist(r, L_1), L_1, far)$
        else
          $Gather()$
        end if
      end if
    else if GatheringSet に含まれていない 3 台のうち，自分が最も L_2 に近い **then**
      Gather()
    end if
  else if $dns = 2$ **then**
    GatheringSet に含まれていない 2 台のうち，自身を r，もう 1 台を r' とする
    if $dist(r, L_3) > dist(r', L_3)$ **then**
      if $dist(r, L_3) < 4$ **then**
        $Slide(2, L_3, far)$
      else
        $Slide(dist(r, L_5), L_5, near)$
      end if
    else
      if $dist(r', L_5) = 0$ **then**
        if $dist(r, L_4) = 0$ **then**
          $Slide(dist(r, L_3) - \sqrt{3}, L_3, near)$
        else if $dist(r, L_3) < 2$ **then**
          $Slide(2, L_3, far)$
        else
          $Slide(dist(r, L_4), L_4, near)$
        end if
      end if
    end if
  else if $dns = 4$ **then**
    $Slide(dist(r, L_3) - \sqrt{3}, L_3, near)$
  end if

図 12 GatheringRobots(R,r)

合でも，L_1 上のロボットは L_1 に垂直に移動するため，L_1 上のロボットが移動した場合でも残り 2 台の L_3 からの距離は異なる．よって，L_3 からの距離を用いて移動するべき 1 台を一意に決定することは可能である．

L_3 に近い方のロボットは，GatheringSet に含まれるロボットに移動を妨げられる可能性があるため，それを防ぐために L_3 に対して垂直方向へ移動する必要がある．しかし，近い方のロボットのみが移動すると，遠い方のロボットとの位置関係が逆転する可能性があるため，位置関係が逆転する可能性がある場合は遠い方のロボットも L_3 から垂直方向に遠ざかる．よって位置関係は逆転しない．

遠い方のロボットが L_5 上へ移動するとき，移動上に障害物は存在しない．同様に近い方のロボットが L_4 上へ移動するときにも障害物は存在しない．また，L_4 上へ移動したロボットが集合しようとするとき，その経路に障害物が存在しないことは明らかである．よって，dns=2 のとき，有限時間内に dns=4 となる． □

補題 3.5. dns=4 のとき，1 台のロボットのみが移動し，有限時間内に dns=7 となる

証明 5. dns=4 のとき，4 台が集合している状態で，最後の 1 台は集合地点まで直線移動するだけで dns=7 の状態で集合を達成する．この経路には障害物は存在せず，その移動距離は有限である．また，その間他のロボットは移動しない．よって，dns=4 のとき，有限時間内に dns=7 となる． □

定理 3.1. アルゴリズムは，有限時間内に 5 台のロボットを $dns = 7$ の状態で集合させる．

証明 6. 補題 3.1，補題 3.2，補題 3.3，補題 3.4，補題 3.5 による． □

4. おわりに

本稿では，連続平面上において大きさを持つ 5 台の自律分散ロボットによる集合問題を解決するアルゴリズムを提案し，その正当性を証明した．

今後の課題として，同モデルにおける $n \geq 6$ の場合や，共通座標系を持たない 5 台以上の不透明なロボットによる集合問題の解決が挙げられる．

参考文献

[1] Alberto Bandettini, Fabio Luiporini, Giovanni Viglietta.: A Survey on Open Problems for Mobile Robots : arXiv:1111.2259v1 [cs.RO] 7 Nov 2011

[2] Taisuke Izumi, Samia Souissi, Yoshiaki Katayama, Nobuhiro Inuzuka, Xavier Defago, Koichi Wada, and Masafumi Yamashita : The Gathering Problem for Two Oblivious Robots with Unreliable Compasses : SIAM J. Comput., 41(1), 26-46

[3] Paola Flocchini, Nicola Santoro, Giovanni Viglietta, Masafumi Yamashita : Rendezvous of Two Robots with Constant Memory : Structural Information and Communication Complexity Lecture Notes in Computer Science Volume 8179, 2013, pp 189-200

[4] Mark Cieliebak, Paola Flocchini, Giuseppe Prencipe, Nicola Santoro : Solving the Robots Gathering Problem : Automata, Languages and Programming, Lecture Notes in Computer Science Volume 2719, 2003, pp 1181-1196

[5] Ando H, Suzuki I, Yamashita M : Distributed Memoryless Point Convergence Algorithm for Mobile Robots with Limited Visibility : Robotics and Automation, IEEE Transactions on (Volume:15 , Issue: 5)

[6] Jurek Czyzowicz, Leszek Gasieniec, Andrzej Pelc : Gathering Few Fat Mobile Robots in the Plane, Theoretical Computer Science 410 (2009) 481-499

[7] Chrysovalandis Agathangelou, Chryssis Georgiou, Marios Mavronicolas : A Distributed Algorithm for Gathering Many Fat Mobile Robots in the Plane, arXiv:1209.3904v1 [cs.DC] 18 Sep 2012

[8] Sruti Gan Chaudhuri and Krishnendu Mukhopadhyaya : Leader Election and Gathering for Asynchronous Transparent Fat Robots without Chirality : arXiv:1208.4484v1 [cs.DC] 22 Aug 2012

[9] Suparno Datta, Ayan Dutta, Sruti Gan Chaudhuri, and Krishnendu Mukhopadhyaya : Circle Formation by Asynchronous Transparent Fat Robots : Distributed Computing and Internet Technology Lecture Notes in Computer Science Volume 7753, 2013, pp 195-207

[10] 伊藤公一, 片山喜章, 和田幸一 : 共通座標系を有するファットロボットのグリッド上での集合について, 第 143 回アルゴリズム研究会, 研究報告アルゴリズム (AL),2013-AL-143(2),1-8(2013-02-22)