

軸の方向に関する共有知識をもたない自律分散ロボット群に対する形状形成アルゴリズム

山中 信岳^{†*} 伊藤 暢浩[†] 片山 喜章[†] 犬塚 信博[†]
 和田 幸一[†]

A Pattern Formation Algorithm for Distributed Autonomous Robots
 without Agreements on Axis Orientations

Nobutake YAMANAKA^{†*}, Nobuhiro ITO[†], Yoshiaki KATAYAMA[†],
 Nobuhiro INUZUKA[†], and Koichi WADA[†]

あらまし 自律分散ロボット群とは、各ロボットが自律的、協調的に動作することにより、全体として一つの問題を解決するロボット群である。これまでに自律分散ロボットを理論的なモデルで抽象化し、一点集合問題、形状形成問題など様々な問題に適用する研究が行われている [1], [2]。本論文で扱う形状形成問題とは、任意の位置に配置されたロボット群が特定の形状を形成する問題である。形状を形成するとは、入力形状を平行移動、回転、拡大縮小、反転させたものとロボットの配置が一致することをいう。形状形成問題に関する研究として [2], [3] ではロボットに与える共通知識の量による、任意の形状に対する形成可能性について議論されている。本研究では、以前の動作で得られた情報を記憶しておくことができないロボットを対象にし、ロボット間で共有する座標系に関する知識が従来よりも少ない場合において任意の形状を形成することができるアルゴリズムを提案し、その正当性を証明する。

キーワード 自律分散ロボット、形状形成問題

1. ま え が き

自律分散システムとはシステム全体を管理する機構をもたず、システムを構成する各要素がそれぞれ自律的、協調的に動作することにより、全体としてシステムの目的を達成するシステムである。そのため、各要素の故障がシステム全体に与える影響を最小限に抑えることが可能である。また、目的に合わせて構成要素を変更することによりシステムを容易に、柔軟に拡張することができる。

自律分散ロボット群に関する研究としては、鈴木、山下らによる [1] や G. Principe らによる [2], [3] などがある。[1] では、自律分散ロボットを理論的に考えるモデルを構築し、形状形成問題を考える上で必要とな

る自律分散ロボットのモデルや形成できる形状について議論されている。しかし、この中ではロボットの動作は一瞬で完了するという仮定があり、完全な非同期なモデルであるとは言い難い。これに対し [2], [3] ではその仮定を取り除いた完全に非同期なロボットのモデルが提案されている。このモデルのロボットを用いた研究として、形状形成問題や一点集合問題がある。

本論文では [2] と同じモデルで自律分散ロボット群による形状形成問題について考える。形状形成問題とは、初期状態として任意の場所に配置されたロボット群が特定の形状を形成する問題である。形状形成問題に関する研究として [2], [3] では各ロボットが座標系に関する知識をどの程度共有していれば任意の形状を形成することができるかについて議論されている。従来の結果を表 1 および図 1 に示す。ここで軸の方向に関する知識を共有するとは、ロボットが独自にもっている座標系における軸がすべてのロボットにおいて平行になり、軸の正負も一致することである。

ここでは、片軸の方向よりも少ない知識を与えた場

[†] 名古屋工業大学, 名古屋市

Department of Computer Science and Engineering, Graduate School of Engineering, Nagoya Institute of Technology, Nagoya-shi, 466-8555 Japan

* 現在 (株) デンソークリエイト

表 1 形状形成問題に関する従来の結果と本論文の結果
Table 1 Previous results for the pattern formation problem.

文献	共通知識	任意の形状を形成できるか
[2]	両軸方向 (図 1(a))	できる
[4]	片軸方向	奇数台 できる
[2]	(図 1(b))	偶数台 できない
本論文	片軸方向	奇数台 できる
[2]		偶数台 できない
[2]	なし (図 1(c))	できない

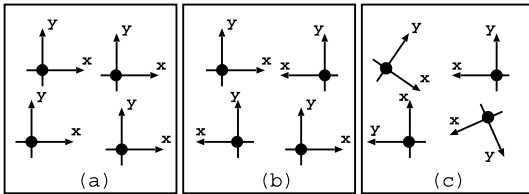


図 1 座標系に関する共通知識
Fig. 1 Agreements on axis coordinate.

合、任意の形状を形成するアルゴリズムが存在するかという問題を考える。軸の方向に関する知識から軸の正負に関する情報を取り除いた軸の方角に注目する。本論文のモデルは以前の動作で得られた情報を記憶しておくことができない。そのため各ロボットが共通の x, y 軸を決定できたとしても、それを動作のたびに決定する必要があり、片軸方向を共通知識として利用する既存のアルゴリズムは利用することができない。

本論文では、奇数台のロボットに片軸の方角を共通知識として与えた場合^(注1)に任意の形状を形成するアルゴリズムを示し、その正当性を証明する。

2. 形状形成問題

2.1 形状形成問題とは

形状形成問題とは、初期状態として任意の位置に配置されたロボットの集合に与えられた形状を形成させる問題である。形状とは点の集合であり各ロボットに同一の形状すなわち点集合が入力され、その数はロボットの台数と等しいとする。各ロボットは初期状態においてすべて異なった位置に配置され、複数のロボットが重なっていることはないとする。

ロボットの集合が形状を形成するとは、ロボットの配置と入力形状を平行移動、反転、回転、拡大縮小させた形状がすべて一致することをいう (図 2)。

入力形状は座標の集合として与えられるため、軸の方向や方角がロボットにより異なる場合、そのまま移動しても形状を形成することはできない。そこで、す

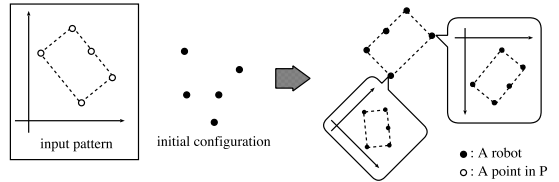


図 2 形状を形成するロボット群
Fig. 2 A pattern formation.

べてのロボット間で共通の目的地を計算するために入力形状に対して平行移動、反転、回転、拡大縮小を行い、その目的地へ移動することにより形状を形成する。

2.2 ロボットのモデル

ここでは、本論文で用いるロボットのモデルを示す。

- すべてのロボットは非同期で動作する。
- 他のロボットを外見から区別することはできない。
- ロボットは体積をもたない点として扱われる。
- 各ロボットは独自の $x-y$ 直交座標系をもつ。詳細は 2.2.1 で述べる。

• ロボットは、観測する任意の時点におけるすべてのロボットの位置を知ることができ、その中で自身の位置を他と区別できる。

- ロボットは通信能力をもたない。
- ロボットは「待機」「観測」「計算」「移動」を行う。これらの動作を 1 サイクルとして、サイクルを繰り返す。詳細は 2.2.2 で述べる。

• ロボットは以前のサイクルで得られた情報を記憶しておくことができない。

- すべてのロボットは同じアルゴリズムを実行する。

2.2.1 ロボットの座標系について

各ロボットがもつ $x-y$ 直交座標系には次の情報が含まれる。

- 軸の方角: x 軸, y 軸として定義する (図 3(a)).
- 軸の方向: 軸の正の側, 負の側として定義する (図 3(b)). どちらか一方の軸の方向が決まることによって、他方の軸の方向が決まることはない。

- 単位距離

• 原点: その座標系をもつロボットの現在位置とする。すべてのロボットが必ずしも同じ座標系をもっているわけではない。そのため、ロボットにより軸の方向、方角、単位距離、原点の位置は異なる可能性がある。

(注1): ただし、各ロボットは $x-y$ 直交座標系を仮定しているため、両軸の方角を知っていることになる。

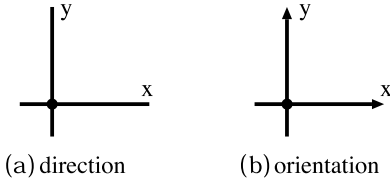


図 3 軸の方角, 方向
Fig. 3 Direction and orientation.

2.2.2 サイクルについて

ロボットの 1 サイクルは次の動作からなる.

- 待機: ロボットはなにも動作しない. 初期状態においてすべてのロボットはこの状態である.
- 観測: 視野に制限のないカメラを用いて各ロボットの位置を観測する. ロボットは観測を開始した瞬間の各ロボットの座標を観測結果として得る.
- 計算: 入力形状, 各ロボットの座標を用いてロボットはアルゴリズムに従って計算し, 現在のサイクルの目的地を得る.
- 移動: 計算によって求められた目的地へ直線で移動する. ただし, そのサイクル内に必ず目的地に到着するとは限らないとする. したがって, 移動できるところまで移動して, 次のサイクルへ移る.

本論文では時間と距離に関して以下の仮定を設ける.
仮定 1 (時間): ロボットが 1 サイクルに必要とする時間は 0 ではない有限時間である. 1 サイクルに少なくとも ϵ の時間が必要であるというロボットに固有の定数 $\epsilon > 0$ が存在する.

仮定 2 (距離): ロボットが 1 サイクルに移動する距離は有限である. 目的地までの距離が δ よりも大きいとき, 目的地へ向かって少なくとも δ だけ移動するというロボットに固有の定数 $\delta > 0$ が存在する.

仮定 1, 2 から, 各行動に必要な時間や移動距離はロボットにより異なるため, 有限であるが予測できない. その結果, 各ロボットには時間に関する共通の概念は存在しない. したがって, 移動中のロボットを観察したり, 実際とは異なるロボットの配置をもとにして目的地を計算する可能性がある.

3. 諸 定 義

本論文中で使われる記号, 言葉を定義する.

- 上下, 左右, 垂直, 水平: x 軸の正の方向を右, 負の方向を左, y 軸の正の方向を上, 負の方向を下と呼ぶ. y 軸に平行であることを垂直, x 軸に平行であることを水平という.

- 垂直距離, 水平距離: 2 点 $(a, b), (c, d)$ が与えられたとき, $|b - d|$ を垂直距離, $|a - c|$ を水平距離という. 2 本の垂直線 (水平線) が与えられたとき, 2 直線の最短距離を水平距離 (垂直距離) という.

- r : ロボット. または, ロボット r がいる点.
- p, f : それぞれ入力形状, 目的地の点.
- $p.x, p.y$: 点 p の x 座標, y 座標.
- R : ロボットの集合, またはロボットがいる点の集合.

- P, F : それぞれ入力形状, 目的地.
- V, H : それぞれ P, F の点を通る垂直線, 水平線.
- W, I : それぞれ R の点を通る垂直線, 水平線.
- \overline{pV} : 点 p と垂直線 V の最短距離.
- $\overline{V_1V_2}$: 垂直線 V_1 と V_2 の最短距離.
- $|V|_P, |W|_R$: それぞれ, V, W 上の P, R の数.
- 辞書式順序: 平面上の点集合 P が与えられたとき, 水平優先辞書式順序 \leq_x を次のように定義する.
 $p_1 \leq_x p_2 \iff$ 「 $p_1.x < p_2.x$ 」または 「 $p_1.x = p_2.x$ かつ $p_1.y \leq p_2.y$ 」 ($p_1, p_2 \in P$)

垂直方向と水平方向を置き換えることで, 同様に垂直優先辞書式順序 \leq_y を定義する.

- 中央点, 中央のロボット: 入力形状の点と奇数台のロボットの集合が与えられたとき, 水平優先辞書式順序における中央の点を $p_{m(x)}$, 中央のロボットを $r_{m(x)}$, それぞれの点を通る垂直線を V_m, W_m と書く. 垂直優先に関しても同様に $p_{m(y)}, r_{m(y)}$, それぞれの点を通る水平線を H_m, I_m とする.

- 差集合: 集合 A, B の差集合 $A \setminus B$ を次のように定義する. $x \in A \setminus B \iff x \in A$ かつ $x \notin B$

- 対称性: 入力形状の点集合 P が与えられたとする. 点 $p \in P$ を通る垂直線を V_p として, V_m に関して対称な組をもつ点集合 P_{symY} を

$$P_{symY} = \{p \in P \mid \exists p' (\neq p) \in P, \overline{pV_m} = \overline{p'V_m} \text{ かつ } |V_p|_P = |V_{p'}|_P \text{ かつ } V_p \neq V_{p'}\}$$

- と定義する. $P \setminus V_m = P_{symY}$ が成り立つとき, P は垂直対称であるという (図 4(a)).

垂直方向と水平方向を置き換えることで, 同様に水平対称を定義する (図 4(b)).

- 非対称な半面: P が垂直対称でないとき, $P \setminus P_{symY}$ の中で V_m からの水平距離が最も大きな点を p_{as} とする. V_m によって分割された二つの半面のうち, より多くの p_{as} を含む半面を y 軸に関して非対称な半面という (図 5(a)). $p_{as} \in P \setminus P_{symY}$ より, V_m によって分割された二つの半面において p_{as} の数

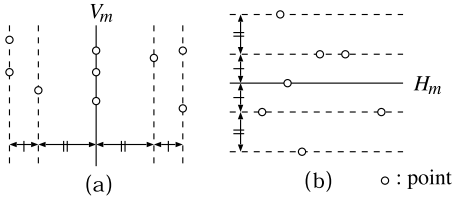


図4 対称性
Fig. 4 Symmetricity.

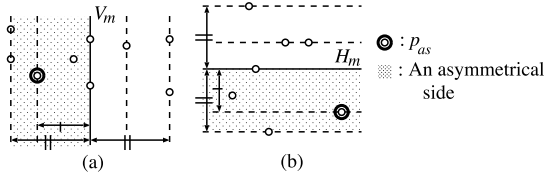


図5 非対称な半面
Fig. 5 An asymmetrical side.

は異なるため、非対称な半面は一意に定まる。

垂直方向と水平方向を置き換えることで、同様に x 軸に関して非対称な半面を定義する(図5(b))。

ロボットの集合に対しても同様に対称性と非対称な半面を定義する。

●点集合 P の凸包: 点集合 P が与えられたとき, P の凸包とは $\forall p \in P$ を含む面積最小の凸多角形をいう。

4. アルゴリズム

4.1 アルゴリズムの概要

各ロボットにおけるアルゴリズムの入力として以下の値を与える。

- 入力形状 P
- 観測したロボットの配置 R

ただしロボットの台数 n は奇数であり, すべてのロボットは y 軸の方角を共通知識として与えられている。

この入力に対して各ロボットは次のようなアルゴリズムをもって動作する。

(1) 基準(線等)と目的地の決定

(a) ある条件を満たすように, 入力形状 P の変換を行い, 入力形状に対する基準(線等)を決定する。

(b) 入力形状と同様の条件を満たすように, 観測したロボットの配置に対して基準を決定する。

(c) P 及び R の基準線を一致させ目的地 F を決定する。

(2) 目的地への移動

(a) 基準を維持できるように移動する。

(b) 衝突しないように目的地 F へ移動する。

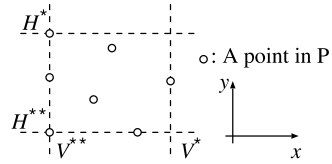


図6 P の凸包接線
Fig. 6 A tangent of P .

次節以降で, まず上述の「ある条件」と「入力形状の変換」について説明した後, このアルゴリズムの詳細について説明する。

4.2 入力形状に対する条件とそれへの変換

各ロボットは入力形状 P を点座標の集合として同一の数値データとして受け取る。入力形状 P が与えられたとき, P の凸包に対する2本の垂直接線のうち, 右側を V^* , 左側を V^{**} とする。また P の凸包に対する2本の水平接線のうち, 上側を H^* , 下側を H^{**} とする(図6)。

また V^*, V^{**} 上で一番上側の点をそれぞれ p_x^*, p_x^{**} とする。

この入力形状を次のような条件を満たすように変換する。

- $|V_m|_P < n - 1$
- $(\overline{H^*H_m} < \overline{H^{**}H_m})$
 $\vee (\overline{H^*H_m} = \overline{H^{**}H_m})$
 $\wedge x$ 軸に対して非対称な半面が H_m の下側にある)
 $\vee (P$ が水平対称)
- $(p_x^*.y > p_x^{**}.y)$
 $\vee (p_x^*.y = p_x^{**}.y \wedge P$ が垂直対称でない
 $\wedge y$ 軸に対して非対称な半面が V_m の左側にある)
 $\vee (p_x^*.y = p_x^{**}.y \wedge P$ が垂直対称)

後述のとおり各ロボットは, R で決まる I_m や W_m に対する非対称な性質から軸の方向を一致させて動作する。上述の条件は, このとき決める軸の方向と入力形状の非対称な性質が一致するための条件である。これが満たされないと, アルゴリズムの動作によって形状が作られるときの軸の方向が反転するなど不都合が生じる。

入力形状の変換は次に示す(1)~(3)の順に行う。

(1) $|V_m|_P \geq n - 1$ の場合

x 軸と y 軸を入れ換え, P を次の形状に置き換える。

$$\{(y, x) | (x, y) \in P\}$$

(2) $\overline{H^*H_m} > \overline{H^{**}H_m}$ の場合

x 軸に対して反転し, P を次の形状に置き換える。

$$\{(x, -y) | (x, y) \in P\}$$

(3) $p_x^* \cdot y < p_x^{**} \cdot y$ の場合

y 軸に対して反転し、 P を次の形状に置き換える。

$$\{(-x, y) | (x, y) \in P\}$$

この変換の結果、上述の条件を満たす。

アルゴリズムでは P がこれらの性質を満たすものとしてロボットの行動を与える。

4.3 アルゴリズムの詳細

4.3.1 基準と目的地の決定

(a) 入力形状 P における、基準となる直線 V_o, V'_o, H_o, H'_o を次の方法で定める。

4.2 のようにして得られる入力形状 P に対して V^{**}, V^*, H^{**}, H^* をそれぞれ V_o, V'_o, H_o, H'_o とし、 V'_o 上で一番上側にある点 p'_x とする。また、 P を垂直優先辞書式で順序づけ、1 番目の点を p_y とする。

(b) ロボットの配置 R において、基準となる直線 W_o, W'_o, I_o, I'_o を次の方法で定める。

ここで入力形状と同様にして、観測されたロボットの配置 R が与えられたとき、 R の凸包に対する 2 本の垂直接線のうち、左側を W^* 、右側を W^{**} とする。また R の凸包に対する 2 本の水平接線のうち、上側を I^* 、下側を I^{**} とする。更に W^*, W^{**} 上で一番上側にある点をそれぞれ r_x^*, r_x^{**} とする。

(1) R の水平接線 I^*, I^{**} を用いて、 I_o, I'_o を定め、 I_o から I'_o の方向を y 軸の正の方向とする。

• $\overline{I^* I_m} \neq \overline{I^{**} I_m}$ である場合

I^*, I^{**} のうち、 I_m からの垂直距離が大きな水平線を I_o とし、もう一方を I'_o とする。

• $\overline{I^* I_m} = \overline{I^{**} I_m}$ かつ水平対称でない場合

x 軸に関して非対称な半面に含まれる水平接線を I_o として、もう一方を I'_o とする。

• R が水平対称の場合 (R, P のいずれかあるいは両方が横一直線である配置は除く)

I^*, I^{**} を決定するができないので、 $r_{m(y)}$ が自身の座標系において上へ垂直に移動することにより対称性をくずし、その進行方向から I_o, I'_o を決定する。 $r_{m(y)}$ は決定した I_o, I'_o により F を計算し、 R, F を通る水平線の中で $r_{m(y)}$ よりも上にあり、 $r_{m(y)}$ から最も近い水平線を I_c とし、 $\overline{I_m I_c}$ 未満の距離を移動する。

• R が水平対称の場合 (R, P がともに横一直線である配置のとき)

I^*, I^{**} を決定するができないが、この場合、ロボットは垂直方向の移動をする必要がなため、それらを決定する必要はない。そこで $I_m = I_o = I'_o$ とし、各ロボットのローカルな座標系で y 軸の正の方向をそのまま y 軸の正の方向とする。

(2) R の垂直接線 W^*, W^{**} を用いて、 W_o, W'_o を定め、 W_o から W'_o の方向を x 軸の正の方向とする。

• $r_x^* \cdot y \neq r_x^{**} \cdot y$ である場合

r_x^*, r_x^{**} のうち上にある点を r'_x として、 r'_x を通る垂直接線を W'_o 、もう一方を W_o とする。

• $r_x^* \cdot y = r_x^{**} \cdot y$ かつ垂直対称でない場合

y 軸に関して非対称な半面に含まれる垂直接線を W_o 、もう一方を W'_o として、 W'_o 上で一番上側にある点 r'_x とする。

• R が垂直対称である場合

$r_{m(x)}$ が右へ水平に移動することにより対称性をくずし、進行方向から W_o, W'_o を決定する。 $r_{m(x)}$ は決定した W_o, W'_o から F を計算し、 R, F を通る垂直接線の中で $r_{m(x)}$ よりも右にあり、 $r_{m(x)}$ からの最も近い垂直接線を W_c とし、 $\overline{W_m W_c}$ 未満の距離を移動する。 $|V_m|_P < n - 1$ より P が縦一直線になることはないため、 W_c は必ず存在する。

(3) R を垂直優先辞書式で順序づけ、1 番目のロボットを r_y とする。

(c) P 及び R の基準から目的地 F を決定する。

V_o と W_o, V'_o と W'_o, H'_o と I'_o を一致させるように P を平行移動、反転、拡大縮小したものを F とする。 p'_x, p_y に相当する F の点を f'_x, f_y と表す。

ここで、 F は W_o と W'_o の間の水平距離を V_o と V'_o の水平距離とみなすことで決まるため、次の場合は F を定めることができない。

• $V_o = V'_o$ の場合

$|V_m|_P < n - 1$ を仮定したので、この場合は起こり得ない。

• $W_o = W'_o$ の場合

$W_o = W'_o$ であれば垂直対称であり、そのとき $r_{m(x)}$ は水平に移動するので、こうしたことはあり得ない。しかし、この $r_{m(x)}$ の移動によって $\overline{W_o W'_o}$ が変化し、観測時刻によって異なることは問題である。そこで W_m 上の中央のロボットを $r_o, d = \overline{r_o I'_o}$ とする。 r_o は W_m から右への一定距離 d 離れた位置へ水平に移動させる (図 7(a))。

また移動中に観測された場合のため $|W_m|_R = n - 1$ の場合は W_m 上にいないロボットを r_o として、 r_o

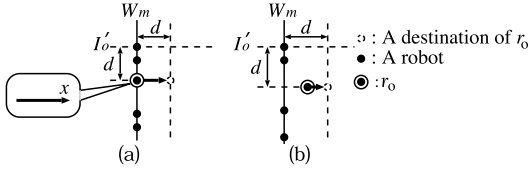


図7 $W_o \neq W'_o$ となるためのロボットの移動
Fig. 7 The movement to configuration $W_o \neq W'_o$.

は $\overline{r_o W_m} < d$ ならば $\overline{r_o W_m} = d$ となる位置まで水平に移動する。その間、他のロボットは待機することで $W_o W'_o$ がロボットごとに異なることのないようにする (図 7 (b))。

4.3.2 目的地への移動

$\mathbf{R} \setminus \{r_y, r'_x\}$ が目的地に到達していないとき、次のようにして目的地へ移動する。

(a) 基準の維持

目的地 \mathbf{F} は前節で定めた基準 (W_o, W'_o, I'_o) より決まるため、ロボットの移動により基準が変化すると \mathbf{F} も変化する可能性がある。そのため、各ロボットは基準線が変わらない配置に移動する。

• I'_o の維持

$\mathbf{R} \setminus \{r_y\}, \mathbf{F} \setminus \{f_y\}$ を通る水平線のうち、 I'_o からの垂直距離が最も大きな水平線を H_{max} とする。 $\overline{I_o H_{max}} \leq \overline{H_{max} I'_o}$ のとき、 r_y は $\overline{r_y H_{max}} = \overline{H_{max} I'_o} + c$ (c は定数) を満たす最短の位置まで垂直に移動する (図 8 (a-1))。 \mathbf{R}, \mathbf{P} とともに横一直線に並んでいる配置では I_o, I'_o, H_{max} は同一直線なので、 I_o から c だけ離れた点は二つ存在する (図 8 (a-2))。そのような場合は、任意の方を選択し移動する。

• W_o, W'_o の維持

r'_x が W'_o と I'_o の交点まで垂直に移動する (図 8 (b))。

ここで、 W_o 上で一番上側にあるロボットを r_x とする。 \mathbf{R} の初期配置として r_x が I'_o 上にある場合、 r_x は $\overline{I'_o I_c}$ 未満の距離を下へ垂直に移動する (図 8 (c))。

(b) 目的地への移動

$\mathbf{R} \setminus \{r_y\}, \mathbf{F} \setminus \{f_y\}$ を水平優先辞書式で並べたとき、 i 番目のロボット、 i 番目の点をそれぞれ r_i, f_i とする ($i = 1, 2, \dots, n-1$)。 r_i は f_i を目的地として、各ロボットは以下の手順で目的地へ移動する。ただし、あるロボットが移動することにより $|W_o|_{\mathbf{R}} = 0$ となる場合、そのロボットは移動しない。

• r_y, r'_x 以外のロボットの場

すべてのロボットがいっせいに目的地への移動を開

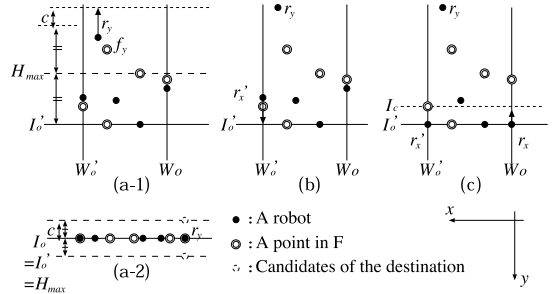


図8 基準線の維持
Fig. 8 The movement to keep base lines.

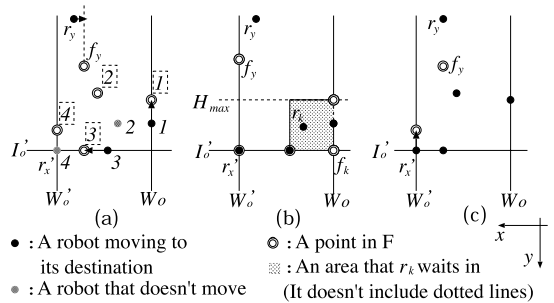


図9 ロボットの移動
Fig. 9 The movement to destination.

始すると、ロボット同士が衝突する、または辞書式順序の変化により目的地が変わる可能性がある。そのため、次の条件を満たすロボット r_i のみ、目的地への移動を開始する (図 9 (a))。

$$\begin{cases} r_i \leq_x f_{i+1} & \text{if } i < n-1 \\ \text{かつ} \\ f_{i-1} \leq_x r_i & \text{if } i > 1 \end{cases}$$

V_o と H'_o の交点に目的地 f_k ($1 \leq k \leq n-2$) が存在する場合、 r_k は辞書式順序を維持しながら、目的地の手前で待機する。つまり、 H_{max} と I'_o の間の領域で

$$\begin{cases} r_1 \leq_x f_2 \text{かつ } r_1 \neq f_1 & \text{if } k = 1 \\ f_{k-1} \leq_x r_k \leq_x f_{k+1} \text{かつ } r_k \neq f_k & \text{if } k > 1 \end{cases}$$

を満たす位置で待機する (図 9 (b))。 r_y, r'_x, r_k 以外のロボットが目的地に到達後、 r_k は f_k へ移動する。

• r'_x 及び r_y の場合

r'_x は $\mathbf{R} \setminus \{r'_x, r_y\}$ が目的地に到着するまで待機する。 r_y は f_y を通る垂直線に向かって水平に移動する (図 9 (a))。

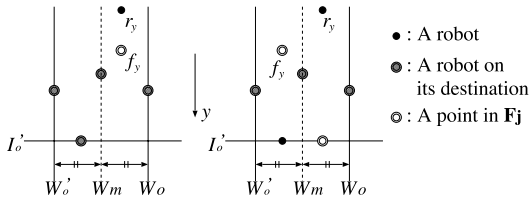


図 10 F が一意に定まらない場合
Fig. 10 The case that F is not unique.

r'_x, r_y 以外のロボットが目的地に到達後, r'_x が目的地に向かって垂直に移動する (図 9 (c)).

r'_x が目的地に達すると, R は次のような配置になる.

- 水平対称でない
- $R \setminus \{r_y\} = F \setminus \{f_y\}$

ここで, R の配置が

- $r_x \cdot y = r'_x \cdot y$
- 垂直対称である

であるとき, W_o, W'_o が一意に定まらないため, すべてのロボットに共通の F を求めることができない. そこで垂直対称な 2 通りの目的地 $F_j (j = 1, 2)$ を想定する (図 10).

最後に r_y が f_y へ向かって移動する. その結果, ロボットの位置と F の点がすべて一致し, 与えられた形状を形成する.

4.3.3 アルゴリズムの正当性

ここではアルゴリズムの正当性を証明する. ロボットの台数 n は奇数であり, 入力形状 P は 4.2 の条件を満たす.

R は次に示す $C_1 \sim C_7$ のうちいずれかに分類される. これを図 11 に示す.

- 状況 C_1 : R が水平対称な配置
- 状況 C_2 : $\{W_o, W'_o\}$ が一意に定まらない配置
- 状況 C_{2-1} : すべてのロボットが同じ垂直線上にいる配置
- 状況 C_{2-2} : それ以外の配置
- 状況 C_3 : 基準線を維持できる配置
- 状況 C_4 : 基準線を維持できない配置
- 状況 C_5 : $R \setminus \{r_y, r'_x\}$ が目的地に到達した配置
- 状況 C_6 : $R \setminus \{r_y\}$ が目的地に到達した配置
- 状況 C_7 : 形状を形成している配置

初期配置が入力形状 (形成すべき形状) と同じであり, かつ R が対称な配置である場合はすべてのロボットが共通の F を一意に求めることができない. そこで入力形状を水平, 垂直反転したものを目的地 $F_1 \sim F_4$ と定義する. R がこのいずれかに一致すれば形状は形

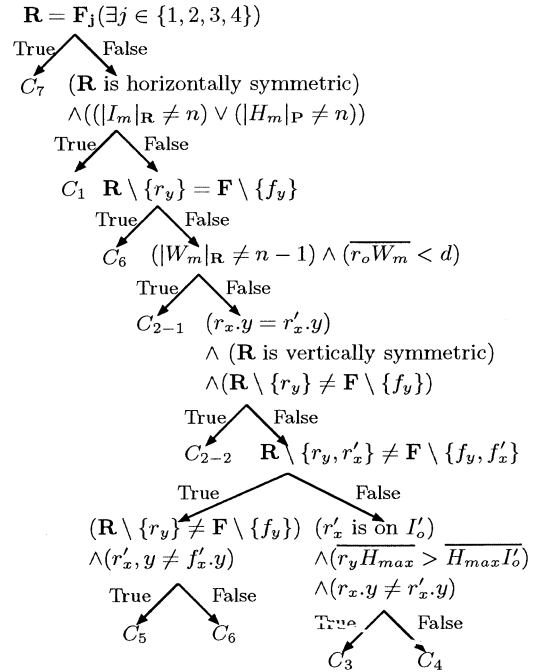


図 11 ロボットの配置 R の分類
Fig. 11 A classification for locations of robots R.

成できる.

初期配置がどの状況であっても, アルゴリズムに従って動作することにより最終的には C_7 になることを示す. C_7 は形状を形成している状況である.

そのためにまず, 次の言葉, 記号を定義する.

- 静止している: 移動していない, かつ現在のサイクルの計算で移動しないという結果が出力されること
- 時刻 t_s : ロボットの配置が状況 C_s から異なる状況に変化する時刻 ($s = 1, 2, \dots, 6$)

これらを用いて, 次の手順でアルゴリズムの正当性を証明する.

1. $C_1 \sim C_6$ すべての状況において, 次に示す条件 (a) を満たしていれば (b), (c) が成り立つことを, 補題 1~7 で示す.

(a) 状況 C_s において, たかだか 1 台のロボットを除くすべてのロボットが静止している.

(b) アルゴリズムの実行により, たかだか 1 台のロボットを除くすべてのロボットが静止する時刻 t_s が存在する. また, 状況の変化は図 12 以外存在しない.

(c) C_s になった瞬間に移動しているロボットが他のロボットの動作に影響を与えることはない.

ここで, 状況 C_s になった瞬間と時刻 t_s において,

表 2 状況 C_s になった瞬間、及び時刻 t_s において移動している可能性があるロボットとその目的地
 Table 2 Robots moving in the moment that it become C_s and t_s , and its destination.

状況	移動している可能性があるロボット		t_s で移動しているロボットの目的地
	C_s になった瞬間	t_s	
C_1	なし	$r_{m(y)}$	$r_{m(y)}$ から上へ $\overline{I_m I_c}$ 未満だけ離れた位置
C_{2-1}	$r_{m(y)}$	$r_{m(y)}$	同上
C_{2-2}	$r_{m(y)}$	$r_{m(x)}$	$r_{m(x)}$ から右へ $\overline{W_m W_c}$ 未満だけ離れた位置
C_4	$r_{m(y)}$ or $r_{m(x)}$	$r_{m(y)}$ or $r_{m(x)}$	$r_{m(y)}$ から上へ $\overline{I_m I_c}$ 未満だけ離れた位置 or
			$r_{m(x)}$ から右へ $\overline{W_m W_c}$ 未満だけ離れた位置
C_3	$r_{m(y)}$ or $r_{m(x)}$	なし	—
C_5	なし	なし	—
C_6	なし	なし	—

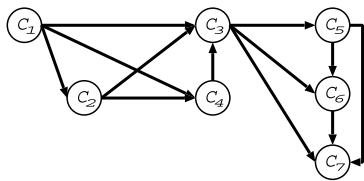


図 12 アルゴリズムにおける状況の変化
 Fig. 12 The state transition diagram in our algorithm.

移動している可能性があるロボットは表 2 に示すようになることが証明できる。

2. 初期配置ではすべてのロボットが静止しているため、1. より、初期配置がどの状況であってもアルゴリズムの実行により図 12 のように状況が変化し、最終的な状況が C_7 になる。

はじめに、 C_1 からの変化について示す。各ロボットは、 R, P とともに横一直線である場合を除いて、形状を形成することなく水平対称な配置に移動することはない。よって、 C_1 は必ず初期配置であり、すべてのロボットは静止している。

[補題 1] $R \in C_1$ であるとする。このとき、アルゴリズムの実行により有限時間内に $R \in C_2 \cup C_3 \cup C_4$ となり、 $r_{m(y)}$ を除くすべてのロボットが静止する時刻 t_1 が存在する。

(証明) $r_{m(y)}$ が移動することにより、対称性をくずす。 R は水平対称な配置なので $r_{m(y)}$ を一意に定めることができる。 $r_{m(y)}$ は $\overline{I_m I_c}$ 未満の距離を垂直に移

動するため他のロボットと衝突することなく、その移動距離は有限である。また、 R, F を通る水平線を越えることはないので、 F の点に到達しない。

$r_{m(y)}$ 以外のロボットは静止しているため、有限時間内に $R \in C_2 \cup C_3 \cup C_4$ となり、 $r_{m(y)}$ を除くすべてのロボットが静止する時刻 t_1 が存在する。□

次に、 C_2 からの変化について示す。ロボットの配置により C_{2-1}, C_{2-2} に分類する。

[補題 2] $R \in C_{2-1}$ において、 $r_{m(y)}$ を除くすべてのロボットが静止しているとき(表 2)、アルゴリズムの実行により有限時間内に $R \in C_3 \cup C_4$ となり、 $r_{m(y)}$ を除くすべてのロボットが静止する時刻 t_2 が存在する。

(証明) 初期配置が C_{2-1} である場合、 r_o が W_m から d だけ離れた位置に水平に移動する。 d は I'_o と r_o の垂直距離により決まり、 r_o は水平に移動するため d は一定である。よって、 r_o の移動距離は有限である。

$R \in C_1$ において、 $r_{m(y)}$ が対称性をくずした瞬間に C_{2-1} になる可能性がある。このとき、 r_o のほかにも $r_{m(y)}$ が移動しているが、 $r_{m(y)}$ は R, F を通る水平線を越えないように垂直に移動するため、水平方向に移動する r_o と衝突しない。

$r_o, r_{m(y)}$ 以外のロボットは静止しているため、 r_o が目的地に到着した瞬間に $r_{m(y)}$ を除くすべてのロボットが静止する時刻 t_2 が存在することになる。□

[補題 3] $R \in C_{2-2}$ において、 $r_{m(y)}$ を除くすべてのロボットが静止しているとき(表 2)、アルゴリズムの実行により有限時間内に $R \in C_3 \cup C_4$ となり、 $r_{m(x)}$ を除くすべてのロボットが静止する時刻 t_2 が存在する。

(証明) 初期配置が C_{2-2} である場合、 $r_{m(x)}$ が移動することより、対称性をくずす。 R は垂直対称なので、 $r_{m(x)}$ を一意に定めることができる。 $r_{m(x)}$ は $\overline{W_m W_c}$ 未満の距離を水平に移動するため他のロボットと衝突することなく、その移動距離は有限である。

$R \in C_1$ において、 $r_{m(y)}$ が対称性をくずした瞬間に C_{2-2} になる可能性がある。このとき、 $r_{m(y)}$ は移動している。 C_1, C_{2-2} はそれぞれ水平、垂直対称な状況なので、 C_{2-2} において $r_{m(y)}$ が移動しているのはロボットの配置が水平対称かつ垂直対称な場合だけである。この場合、 $r_{m(y)}$ と C_{2-2} において移動を始める $r_{m(x)}$ は同じロボットである。よって、 $r_{m(y)}$ の動作が他のロボットに影響を与えることはない。

$r_{m(x)}$ 以外のロボットは静止しているため、有限時

間に $\mathbf{R} \in C_3 \cup C_4$ となり, $r_{m(x)}$ を除くすべてのロボットが静止する時刻 t_2 が存在することになる。□

[補題 4] $\mathbf{R} \in C_4$ において, $r_{m(y)}, r_{m(x)}$ のどちらか 1 台を除くすべてのロボットが静止しているとき (表 2), アルゴリズムの実行により有限時間内に $\mathbf{R} \in C_3$ となり, 表 2 に示すロボットのうちどちらか 1 台を除くすべてのロボットが静止する時刻 t_4 が存在する。

(証明) 初期配置が C_4 である場合, r_x, r'_x, r_y が移動する。 r_y は I'_o からの垂直距離が最大のロボットなので, 進行方向に他のロボットは存在しない。また, I'_o から離れて移動するので r_y であることは変わらない。 r'_x は W'_o で最も上にいるロボットなので, 進行方向に他のロボットは存在しない。

初期配置として r_x が I'_o 上にいる場合, r_x は移動するが, \mathbf{R}, \mathbf{F} を通る水平線を越えないように垂直に移動するため, W_o が変化することはない。以上のことから, 移動を開始したロボットが他のロボットと衝突することはない, その移動距離は有限である。

$\mathbf{R} \in C_1, C_{2-2}$ において, それぞれ $r_{m(y)}, r_{m(x)}$ が対称性をくずし, その後 C_4 になる可能性がある。このとき, $r_{m(y)}, r_{m(x)}$ のどちらか一方が移動している。 $r_{m(y)}$ である場合, W_o と W'_o の間で移動するため, W_o, W'_o 上を移動する r_x, r'_x と同じ垂直線上にいることはない。 $r_{m(x)}$ である場合, \mathbf{R} を通る垂直線を越えないように水平に移動するため, 垂直方向に移動する r_x, r'_x や, 反対方向に移動する r_y と衝突することはない。よって, これらのロボットの移動が他のロボットの動作に影響を与えることはない。

他のロボットは静止しているため, r_x, r'_x, r_y が目的地に到着した瞬間に $r_{m(y)}, r_{m(x)}$ のどちらか 1 台を除くすべてのロボットが静止する時刻 t_4 が存在することになる。 □

C_3 からの変化を証明するために補題 a~c を示す。

[補題 a] $\mathbf{R} \in C_3$ において, $r_{m(y)}, r_{m(x)}$ のどちらか 1 台を除くすべてのロボットが静止しているとき (表 2), アルゴリズムを実行しても \mathbf{F} が変化することはない。

(証明) \mathbf{F} の形状を維持するために, ロボットが移動しても基準線 W_o, W'_o, I'_o は変化しないことを示す。

r_x, r'_x のうち I'_o 上に存在するのは r'_x だけなので, 常に $r_{x,y} < r'_{x,y}$ が成り立つ。 f_k を目的地とするロボット r_k は $\mathbf{R} \setminus \{r_y, r'_x, r_k\}$ が目的地に到達するまで, 4.3.2 で示した条件を満たす位置で待機する。ま

た, あるロボットが移動することにより $|W_o|_{\mathbf{R}} = 0$ となる場合, そのロボットが移動することはない。よって, W_o, W'_o は変化しない。

r_y は水平方向に移動するため, $\overline{I_o H_{max}} > \overline{H_{max} I'_o}$ は維持される。よって, I'_o は変化しない。

以上のことから, \mathbf{F} が変化することはない。 □

[補題 b] $\mathbf{R} \in C_3$ において, 移動するための条件を満たすロボットが, 少なくとも 1 台は存在する。

(証明) $\mathbf{R} \setminus \{r_y\}$ のロボットの台数 $n - 1$ について帰納法で証明する。

- 1 台のとき, そのロボットが条件を満たす。

- l 台のとき, 条件を満たすロボットが少なくとも 1 台は存在すると仮定し, r_{l+1} を除いたロボットについて考える。その場合, 仮定より l 台のロボットのうち少なくとも 1 台は条件を満たす。

r_{l+1} を考慮したとき, 条件を満たすロボットを r_i とすると, $i \neq l$ のとき, r_i が移動する条件に r_{l+1} は関与しない。よって, r_i は条件を満たしたままである。 $i = l$ のとき, $r_{l+1}(= r_{n-1})$ は必ず I'_o 上にいることから $f_l \leq_x f_{l+1} \leq_x r_{l+1}$ が成り立ち, r_l が移動することができる。よって, $l + 1$ 台のときも条件を満たすロボットが少なくとも 1 台は存在する。 □

[補題 c] $\mathbf{R} \in C_3$ において, アルゴリズムに従って動作しても $\mathbf{R} \setminus \{r_y\}$ の順序は変化しない。

(証明) r_y は辞書式に順序づけられていないので, $\mathbf{R} \setminus \{r_y\}$ について考える。

辞書式順序で隣り合うロボット r_i, r_{i+1} について, どちらかのロボットが停止している場合, もう一方のロボットが目的地に移動しても明らかに順序は変化しない。両方とも移動している場合, $r_i \leq_x f_{i+1}$ かつ $f_i \leq_x r_{i+1}$ かつ $f_i \leq_x f_{i+1}$ が成り立つため, ともに目的地に移動したとしても順序は変化しない。これは辞書式順序で隣り合う任意の 2 台について成り立つ。 □

[補題 5] $\mathbf{R} \in C_3$ において, $r_{m(y)}, r_{m(x)}$ のどちらか 1 台を除くすべてのロボットが静止しているとき (表 2), アルゴリズムの実行により有限時間内に $\mathbf{R} \in C_5 \cup C_6 \cup C_7$ となり, すべてのロボットが静止する時刻 t_3 が存在する。

(証明) 初期配置が C_3 である場合, $\mathbf{R} \setminus \{r_y, r'_x\}$ と r_y の移動方法はことなるため, それぞれの場合に分けて考える。 r'_x は移動しない。

$\mathbf{R} \setminus \{r_y, r'_x\}$ は目的地へ移動する。補題 a より, 移動している間に \mathbf{F} が変化することはない。補題 b

より、移動することのできるロボットが常に存在する。補題 c より、ロボットが移動することにより辞書式順序が変化することはないので、目的地が変化することはない。F は変化しないためロボットの移動距離は有限であり、ロボットの数も有限であることから、各ロボットは有限時間内に目的地に到着する。

r_y は f_y を通る垂直線に向かって水平に移動する。 r_y は I'_o からの垂直距離が最大のロボットなので、 r_y を通る水平線上に他のロボットは存在しない。

$R \in C_1, C_{2-2}$ において、それぞれ $r_{m(y)}, r_{m(x)}$ が対称性をくずし、その後 C_3 になる可能性がある。このとき、 $r_{m(y)}, r_{m(x)}$ のどちらか一方が移動しているが、これらのロボットは R, F を通る直線を越えないように移動するため、辞書式順序が変化することはない。つまり、他のロボットと衝突することはない。

r'_x 及び目的地に到着したロボットは静止しているため、最後のロボットが目的地に到着した瞬間にすべてのロボットが静止する時刻 t_3 が存在することになる。

[補題 6] $R \in C_5$ において、すべてのロボットが静止しているとき、アルゴリズムの実行により有限時間内に $R \in C_6 \cup C_7$ となりすべてのロボットが静止する時刻 t_5 が存在する。

(証明) r'_x が f'_x へ移動する。 r'_x は W'_o 上を移動し、目的地との間にロボットは存在しないため、 r'_x は有限時間内に f'_x に到達し、 $R \in C_6 \cup C_7$ となる。他のロボットは静止しているため、 r'_x が到着した瞬間にすべてのロボットが静止する時刻 t_5 が存在することになる。

[補題 7] $R \in C_6$ において、すべてのロボットが静止しているとき、アルゴリズムの実行により有限時間内に $R \in C_7$ となり、すべてのロボットが静止する時刻 t_6 が存在する。

(証明) r_y が f_y に向かって移動する。 r_y が目的地に到達した瞬間の x, y 軸の方向について考える。

• x 軸の方向について、「 $r_x \cdot y = r'_x \cdot y$ 」かつ「R が垂直対称である」場合、F の候補として垂直対称な 2 通りの目的地 F_j を計算し、 $R = F_j$ である形状を F とする。それ以外の場合、既に目的地に到達している r_x, r'_x は変わらないため、 x 軸の方向は変化しない。

• y 軸の方向について、R が水平対称な場合、F の候補として水平、垂直対称な 4 通りの目的地 F_j を計算し、 $R = F_j$ である形状を F とする。それ以外の場合、 r_y が f_y に到達しても r_y であるロボットは変

らないため、 y 軸の方向は変化しない。

x, y 軸の方向は変わらないため、F は変化しない。よって、各ロボットは F を形成した後、待機する。

他のロボットは静止しているため、 r_y が目的地に到着した瞬間にすべてのロボットが静止する時刻 t_6 が存在することになる。

[定理 1] 奇数台のロボットは有限時間内に任意の形状を形成する。

5. む す び

本研究では、一つの軸の方角を共通知識として与えられた奇数台の自律ロボット群が任意の形状を形成するアルゴリズムを示し、その正当性を証明した。

今後の課題として、本研究とは異なる共通知識をロボット群に与えた場合、任意の形状を形成することができるのか、できなければどのような形状であれば形成することができるのかを示すことが挙げられる。

文 献

- [1] I. Suzuki and M. Yamashita, "Distributed anonymous mobile robots—Formation and agreement problems," 3rd International Colloquium on Structural Information and Communication Complexity (SIROCCO 96), pp.313–330, June 1996.
- [2] G. Prencipe, Distributed Coordination of a Set of Autonomous Mobile Robots, PhD Thesis, Università di Pisa, 2002.
- [3] G. Prencipe, "Corda: Distributed coordination of a set of autonomous mobile robots," 4th European Research Seminar on Advances in Distributed Systems (ERSADS 2001), pp.185–190, May 2001.
- [4] 糟谷政夫, 伊藤暢浩, 犬塚信博, 和田幸一, "片軸方向の共通知識をもつ自律分散ロボット群に対する形状形成アルゴリズム," 信学論 (D-I), vol.J87-D-I, no.7, pp.747–757, July 2004.

付 録

アルゴリズムの詳細

formation(P, R) % P: 与えられた形状 (点集合),

% R: 観測したロボットの座標点集合

normalize(P); $p_x, p_y, H_o, H'_o, V_m = P$ の基準;

$W_m = x$ 軸に関する中央のロボットを通る垂直線;

$I_m = y$ 軸に関する中央のロボットを通る水平線;

$W^*, W^{**} = R$ の垂直接線;

$I^*, I^{**} = R$ の水平接線;

case classify(R) of

C_1 : % R が x 軸に関して対称な配置

$r_{m(y)} = I_m$ 上の中央のロボット;

if 自分は $r_{m(y)}$ then

自身の座標系の上側を進行方向とする;

進行方向から I'_o を決定する;

$W^* = V_o, W^{**} = V'_o, I'_o$ を H'_o と一致させ,

F を通る水平線を計算する;

```

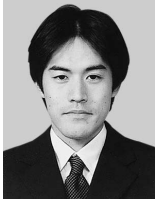
 $I_c = R$  と  $F$  を通る水平線で  $I_m$  よりも上
    があり,  $I_m$  から最近の水平線;
 $\overline{I_m I_c}$  未満の任意の距離だけ垂直に移動する
else 動かない
C2: %  $W_o, W'_o$  が一意に定まらない配置
 $r_{m(x)} = W_m$  上の中央のロボット;
if  $|W_m|_R = n$  then %  $d$  の値を決める
     $r_o = r_{m(x)}$ ;  $d = \overline{r_o I'_o}$ 
elseif  $|W_m|_R = n - 1$  then
     $r_o = W_m$  上でないロボット;  $d = \overline{r_o I'_o}$ 
if  $|W_m|_R \geq n - 1 \wedge \overline{r_o W_m} < d$  then
    %  $R \in C_{2-1}$ 
    if 自分は  $r_o$  then
         $W_m$  から  $d$  離れた位置へ水平に移動する
    else 動かない
else %  $R \in C_{2-2}$ 
    if 自分は  $r_{m(x)}$  then
        自身の座標系の右側を進行方向とする;
        進行方向から  $W_o, W'_o$  を決定する;
         $W_o$  を  $V_o, W'_o$  を  $V'_o, I'_o$  を  $H'_o$  と一致させ,
         $F$  を通る垂直線を計算する;
         $W_c = R$  と  $F$  を通る垂直線の中で  $W_m$  より
        も右にあり,  $W_m$  から最近の垂直線;
         $\overline{W_m W_c}$  未満の任意の距離を水平に移動する
    else 動かない
C3: % 基準線を維持できる配置
 $i = R \setminus \{r_y\}$  の  $\leq_x$  での順番 ( $i = 1, \dots, n - 1$ )
 $r_i =$  自分の位置;
 $f_i = F \setminus \{f_y\}$  における  $\leq_x$  での  $i$  番目の目的地;
 $f_k = V_o$  と  $I'_o$  の交点にある目的地;
if 自分は  $r_y$  then
    if 自分は  $W_o$  上の唯一のロボット then 動かない
    else  $f_y$  を通る垂直線へ水平に移動する
elseif  $r_i = f_i \vee$  自分は  $r'_x (= r_{n-1})$  then 動かない
elseif  $i = 1$  then
    if (自分は  $W_o$  上の唯一のロボット
         $\wedge$  目的地  $f_i$  が  $W_o$  上でない) then 動かない
    elseif  $f_i \leq_x r_{i+1}$  then
        if 自分は  $r_k \wedge$ 
             $R \setminus \{r_k, r'_x, r_y\} \neq F \setminus \{f_k, f'_x, f_y\}$  then
             $f_i \leq_x r_i \leq_x f_{i+1} \wedge r_i \neq f_i$  を満たす任意の
            位置に移動
        else  $f_i$  に移動
    else 動かない
elseif  $r_{i-1} \leq_x f_i \wedge f_i \leq_x r_{i+1}$  then
    if 自分は  $r_k \wedge R \setminus \{r_k, r'_x, r_y\} \neq F \setminus \{f_k, f'_x, f_y\}$  then
         $f_{i-1} \leq_x r_i \leq_x f_{i+1} \wedge r_i \neq f_i$  を満たす任意の
        位置に移動する
    else  $f_i$  に移動する
else 動かない
C4: % 基準線を維持できない配置
if 自分は  $r_y \wedge \overline{I'_o H_{max}} \geq \overline{H_{max} r_y}$  then
     $H_{max} r_y = \overline{I'_o H_{max}} + c$  ( $c$  は定数) を満たす最短位置
    に垂直に移動 (複数存在する場合は任意に選択)
elseif 自分は  $r'_x \wedge I'_o$  上にいない then
     $I'_o$  まで垂直に移動する
elseif 自分は  $r_x \wedge I'_o$  上にいる then
     $I_c = R$  と  $F$  を通る水平線の中で  $I'_o$  よりも下にあり,
     $I'_o$  に最近の水平線;
     $\overline{I'_o I_c}$  未満の任意の距離を垂直に下へ移動
else 移動しない
C5: %  $r_y, r'_x$  以外のロボットが目的地に到達した配置
if 自分は  $r'_x$  then  $f_x (= f_{n-1})$  へ垂直に移動
else 動かない
C6: %  $r_y$  以外のロボットが目的地に到達した配置
if 自分は  $r_y$  then  $f_y$  へ移動 else 動かない
C7: 動かない. % すべてが目的地に到達した配置

normalize(P) % P が条件を満たさない場合, 変換する
 $V_m = \leq_x$  で P の中央の点を通る垂直線;
if  $|V_m|_P > n - 2$  then  $P = \{(y, x) \mid (x, y) \in P\}$ ;
 $H_m = \leq_y$  で P の中央の点を通る水平線;
 $H^* = P$  の凸包に接する下の水平接線;
 $H^{**} = P$  の凸包に接する上の水平接線;
if  $\overline{H^{**} H_m} > \overline{H^* H_m} \vee (\overline{H^{**} H_m} = \overline{H^* H_m}$ 
     $\wedge P$  の非対称な半面が上側)
    then  $P = \{(x, -y) \mid (x, y) \in P\}$ ;
 $p_y^* = P$  の凸包に接する左の鉛直線上の最上部の点の  $y$  座標;
 $p_y^{**} = P$  の凸包に接する右の鉛直線上の最上部の点の  $y$  座標;
if  $p_y^{**} > p_y^* \vee (p_y^{**} = p_y^* \wedge P$  の非対称な半面が右側)
    then  $P = \{(-x, y) \mid (x, y) \in P\}$ ;

classify(R) % R を C1 ~ C7 に分類する
if R は x 軸対称  $\wedge (|I_m|_R \neq n \vee |H_m|_P \neq n)$  then
    if R は P を並行移動, 拡大縮小, 反転したものと一致 then
        return C7
    else return C1
else
    R の基準  $I'_o, r_x, r'_x$  を計算する;
     $I_o \rightarrow I'_o$  を  $y$  軸の正の方向とする;
    if  $r_x.y = r'_x.y \wedge R$  は  $y$  軸対称 then
        R の基準  $W_o, W'_o$  を計算;
         $F_{j'} = H'_o$  と  $I'_o, (V_o, V'_o)$  と  $(W^*, W^{**})$  が一致する
        ように P を並行移動, 拡大縮小, 反転したものと
        ( $j' = 1, 2$ );
        for  $t = 1$  to 2 do
            if  $R = F_t$  then return C7;
             $r_y = \leq_y$  で 1 番目の R のロボット;
             $f_y = \leq_y$  で 1 番目の  $F_t$  の点;
            if  $R \setminus \{r_y\} = F_t \setminus \{f_y\}$  then
                 $F = F_t$ ;
                return C6 %  $y$  軸対称でも,  $r_y$  以外のロボットが
                形状を形成していれば C6
        return C2
    else % F が一意に定まる配置
        R の基準  $W_o, W'_o$  を計算;
         $W_o \rightarrow W'_o$  を  $x$  軸の正の方向とする;
         $F = V_o$  と  $W_o, V'_o$  と  $W'_o, H'_o$  と  $I'_o$  が一致するように
        P を並行移動, 拡大縮小, 反転したものと;
         $r_y = \leq_y$  で 1 番目の R のロボット;
         $f_y = \leq_y$  で 1 番目の  $F_t$  の点;
         $f'_x = P$  における  $p'_x$  に相当する F の点;
         $H_{max} = R \setminus \{r_y\}, F \setminus \{f_y\}$  を通る水平線のうち,  $I_o$  から
        の垂直距離が最も小さい水平線;
        if  $R = F$  then return C7
        elseif  $R \setminus \{r_y\} = F \setminus \{f_y\}$  then
            return C6
        elseif  $R \setminus \{r_y, r'_x\} = F \setminus \{f_y, f'_x\}$  then
            return C5
        elseif  $r'_x$  が  $I'_o$  上にいる
             $\wedge \overline{r_y H_{max}} > \overline{H_{max} I'_o} \wedge r_x.y \neq r'_x.y$  then
            return C3
        else return C4.

```

(平成 16 年 3 月 26 日受付, 8 月 23 日再受付)



山中 信岳 (学生員)

2002 名工大・電気情報工卒。2004 同大大学院博士前期課程電気情報工学専攻了。同年4月(株)デンソークリエイイト入社、現在に至る。自律分散型ロボットに関する研究に興味をもつ。



伊藤 暢浩 (正員)

1994 名工大・工・電気情報工卒。1996 同大大学院博士前期課程了。1999 同大大学院博士後期課程了。博士(工学)。同年4月同大工学部助手。現在に至る。オブジェクト指向技術及びエージェント指向技術に関する研究に従事。情報処理学会, 人工知能学会, IEEE 各会員。



片山 喜章 (正員)

平2 阪大・基礎工・情報卒。平6 同大大学院博士後期課程中退。同年奈良先端科学技術大学院大学情報科学研究科助手。平7 同大情報科学センター助手。平14 名工大工学部講師。平15 同大大学院講師。現在に至る。分散プロトコル, ネットワーク診断などに関する研究に従事。博士(工学)。情報処理学会会員。



犬塚 信博 (正員)

1987 名工大・工・情報工卒。1992 同大大学院工学研究科博士課程電気情報工学専攻了。博士(工学)。同年, 同大工学部助手。2000 同大助教授。現在に至る。1994より1996まで英国インペリアルカレッジを訪問。人工知能, 特に機械学習, 知識表現に関する研究に従事。情報処理学会, 人工知能学会, ソフトウェア科学会, ACM, IEEE Computer, AAAI 各会員。



和田 幸一 (正員)

1978 阪大・基礎工・情報卒。1983 同大大学院博士後期課程了。同年同大・基礎工・助手。1984 名工大・情報(現電気情報)講師。現在教授。工博。1987~1988 ミネソタ大学, ウィスコンシン大学客員助教授。2000 アーヘン工科大学客員教授。VLSIの複雑さ, 並列分散アルゴリズム, グラフ理論の研究に従事。情報処理学会, ソフトウェア科学会, 日本応用数理学会, ACM, IEEE, EATCS 各会員。