

# 自律分散型ロボット群による警邏問題

宇佐美 智之<sup>2,a)</sup> 山内 由紀子<sup>2,b)</sup>

**概要:** 本稿では、二次元空間中を移動するロボット群による警邏問題を考える。ロボット群による警邏問題とは、ロボットが与えられた訪問点の集合を繰り返し訪問する移動スケジュールを求める問題である。各ロボットは匿名かつ無記憶で同期して動作する。本稿では、訪問点が円周上に存在する場合に、訪問点を訪問する間隔が最小となる警邏スケジュールを与える分散アルゴリズムを提案する。また、最適な警邏ができない特別な場合に漸近的に最適な警邏スケジュールを表現する分散アルゴリズムを与える。

TOMOYUKI USAMI<sup>2,a)</sup> YUKIKO YAMAUCHI<sup>2,b)</sup>

## 1. はじめに

モバイルロボットやドローンの普及により、多数の移動計算主体の分散協調が注目されている。分散計算理論では、モバイルロボット、エージェント、プログラマブルな粒子群、変形可能ロボットといった多数の計算モデルとともに、リーダー選挙問題、集合問題、形状形成問題、探索問題、避難問題といった問題が研究されてきた [1]。計算主体群から単一のリーダーを選ぶリーダー選挙問題、移動計算主体群を一点に集合させる集合問題、移動計算主体群に指定した形状を形成させる形状形成問題に共通するのは、移動計算主体群での合意形成である。一方で、探索問題や避難問題では移動計算主体群が効率よく探索範囲などを分担し、協調することが必要となる。

本稿では、モバイルロボット群による警邏問題を提案し、ある条件を満たす初期配置から効率よく警邏問題を解くための分散アルゴリズムを提案する。警邏問題とは、与えられた訪問点の集合をモバイルロボット群で繰り返し訪問する問題であり、本稿では各訪問点をいずれかのロボットが訪問する間隔であるアイドルタイムを最小化することを目指す。警邏問題を解く集中型アルゴリズムはこれまでに多数報告されているが、分散アルゴリズムについてはほとんど研究されてこなかった [1]。

ロボットは識別不能 (匿名) でメモリを持たず (無記憶)、大域座標系を知らず、通信能力を持たない。各ロボットは

観測、計算、移動から成るサイクルを繰り返し実行する。観測フェーズでは他のロボットの配置を自身の局所座標系で観測し、計算フェーズでは共通のアルゴリズムに従って移動経路を計算し、移動フェーズでは計算した移動経路に従って移動する。モバイルロボット群に指定したパターンを形成させるパターン形成問題については既に多くの研究結果が報告されており、ロボット群の初期配置がもつ回転対称性から形成可能なパターンが決まることや、無記憶性、非同期性の影響が報告されている [2-4]。本研究では訪問点が円周上に配置されている時、訪問点の配置の対称性とロボットの台数をもとに、アイドルタイムの最小化を検討する。

## 2. 準備

### 2.1 モデル

ロボットは二次元空間上の点であり、 $n$  台のロボットの集合を  $R = \{r_1, r_2, \dots, r_n\}$  とする。ロボットは識別不能 (匿名) であるが、記述のために  $r_i$  という記法を用いる。各  $r_i$  の大域座標系での座標を  $(x_i, y_i)$  と表し、ロボットの配置を  $P = \{(x_i, y_i) \mid r_i \in R\}$  とする。大域座標系を  $Z_0$ 、ロボット  $r_i$  の局所座標系を  $Z_i$  とする。  $Z_i$  は常に  $r_i$  の現在位置が原点で、 $x$  軸、 $y$  軸の向き、単位長は任意であるが、アルゴリズムの実行中は不変であるとする。全てのロボットは時計回りに合意しているとする。

ロボットは観測、計算、移動の3つのフェーズからなるサイクルを繰り返し実行する。観測フェーズでは自身の局所座標系で他のロボットの位置を観測する。計算フェーズでは観測結果をもとに、共通のアルゴリズムを用いて移動

<sup>1</sup> 九州大学 工学部 電気情報工学科

<sup>2</sup> 九州大学 大学院システム情報科学研究院

<sup>a)</sup> usami.tomoyuki.826@s.kyushu-u.ac.jp

<sup>b)</sup> yamauchi@inf.kyushu-u.ac.jp

経路を決定する。その結果をもとに移動フェーズで平面上を移動する。このときの移動速度はすべてのロボットで等しく、1 サイクルでの最大移動距離を 1 とする。ロボットは最大移動距離を知らないが、計算フェーズで決定した移動経路の長さが 1 より大きい場合、移動フェーズで最大移動距離を移動すると停止する。時間については離散時間  $t = 0, 1, 2, \dots$  を考える。すべてのロボットは毎時刻サイクル、各フェーズを同期して行う完全同期 (FSYNC) スケジューラに従う。各時刻  $t$  にロボットが存在する地点の集合をその時刻の配置とする。

点の集合  $P$  のすべての点を含む最も直径の小さい円を最小包含円といい、 $P$  の最小包含円の中心を点  $c(P)$  と表す。点の集合  $P$  が与えられたとき、以下の条件を満たす最大の自然数  $q$  を  $P$  の対称度  $\rho(P)$  という [3]:  $P$  の最小包含円内部をその中心  $c(P)$  を中心とするちょうど  $q$  個の扇形の領域に分けることができ (各扇形の内角は  $2\pi/q$ )、かつ、最小包含円内部の点の集合を  $x_1, x_2, \dots, x_q$  とすると、 $x_i + 1$  が  $x_i$  を最小包含円の中心に対し  $2\pi/q$  だけ回転させたものに一致する。

## 2.2 警邏問題

本稿では、二次元平面上の訪問点の集合とそれらを巡視するロボット群が与えられたとき、ロボット群が訪問点を繰り返し訪問する移動スケジュールを求める警邏問題を考える。

$m$  個の訪問地点の集合を  $Q = \{q_1, q_2, \dots, q_m\}$  とする。訪問点は識別不能であり、記述のために  $q_i$  という記法を用いる。各ロボットは観測フェーズで他のロボットと訪問点を自身の局所座標系で観測する。各ロボットはロボットが存在する点と訪問点を区別できる。ロボットが訪問点を通じたとき、その訪問点を訪問したものとする。ある訪問点について、あるロボットが訪問してから次にいずれかのロボットが訪問するまでの時間をアイドルタイムと呼ぶ。このとき訪問するロボットは同一のロボットでも異なるロボットでもよい。ロボットの移動スケジュール  $S$  が与えられたとき、最大のアイドルタイムを  $T_S$  とする。

本稿では、訪問点が半径  $r (r \geq 1)$  の円周上に存在し、ロボットの台数が訪問点の対称度の約数であるとする。

## 2.3 中心ビュー

本稿で考える中心ビュー (centred view) は、[5] で提案された中心ビューを、訪問点も含むように拡張したものである。

時刻  $t$  におけるロボット  $r_i$  の中心ビュー  $CV_{r_i}(t)$  は、ロボット  $r_i$  の局所座標系と観測結果をもとに、以下のような座標系  $Z_{r_i,t}$  を定め、 $Z_{r_i,t}$  で観測したロボットと訪問点を表す 3 項組の  $CV_{r_i}(t) = \{(x_i, y_i, k)\}$  である。ここで、 $k = 0$  ならば  $(x_i, y_i)$  にロボットが存在し、 $k = 1$  ならば

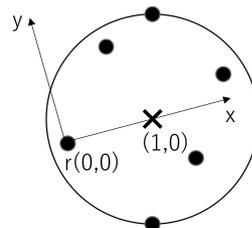


図 1 中心ビューの例。ロボットを黒い点で表し、 $Z_{r_i,t}$  を示す。

$(x_i, y_i)$  は訪問点である。ここでの座標系  $Z_{r_i,t}$  は以下のように定める:  $P$  を現在の配置とする。ロボット  $r_i$  が  $P$  の最小包含円の中心にない場合、 $r_i$  自身の現在地を原点  $(0, 0)$  とし、最小包含円の中心の点を  $(1, 0)$  と定める。ロボット  $r_i$  が  $P$  の最小包含円の中心にいる場合、 $r_i$  自身の現在地を原点  $(0, 0)$  とし、他のロボットと訪問点の中で最小の中心ビューをもつようなロボットの存在する点、または訪問点を  $(1, 0)$  とする (図 1)。定義より、各ロボットが自身の観測結果から、他のロボット  $r_j$  の  $Z_{r_j,t}$  と  $CV_{r_j}(t)$  を計算できる。

## 3. 提案アルゴリズム

本章では同期スケジューラの下での無記憶なロボット群による警邏アルゴリズムを与える。

### 3.1 セクター

訪問点の集合  $Q$  が与えられたとき、 $Q$  の最小包含円の中心を端点とする 2 本の半直線  $l_{S_i}^s, l_{S_i}^f$  で囲まれた領域が以下の条件を満たすならば、その領域をセクター  $S_i$  と呼ぶ。ただし、 $S_i$  は  $i$  番目 ( $i = 1, 2, \dots, \rho(Q)$ ) のセクターを表し、反時計回りに半直線  $l_{S_i}^s, l_{S_i}^f$  と並ぶ。また、2 本の半直線  $l_{S_i}^s, l_{S_i}^f$  のなす角を  $\theta_i$  と表す。

- (1)  $\theta_1 = \theta_2 = \dots = \theta_{\rho(P)} < 2\pi/\rho(P)$ .
- (2) 半直線  $l_{S_i}^s, l_{S_i}^f$  上に  $P$  の点を含む (それぞれの点を  $p_{S_i}^s, p_{S_i}^f$  とする)。
- (3) セクター  $S_i$  に含まれる点を  $2\pi/\rho(P)$  だけ反時計回りに回転させると  $S_{i+1(\text{mod } \rho(P))}$  に含まれる点に重なる ( $i = 1, 2, \dots, \rho(P)$ )。
- (4) セクターのなす角が最小になるように半直線  $l_{S_i}^s, l_{S_i}^f$  を選ぶ。ただし、半直線  $l_{S_i}^s, l_{S_i}^f$  の選び方が複数通り存在する場合、その中で中心ビューが最小となる点を含む半直線  $l_{S_i}^s$  を選ぶ。

定義より、以下の補題が成立する。

**補題 1** 訪問点の集合  $Q$  が与えられたとき、それらに分けるセクターは一意に決まる。

### 3.2 提案アルゴリズム

本稿では、ロボットの台数  $n = |P|$  は訪問点の対称度

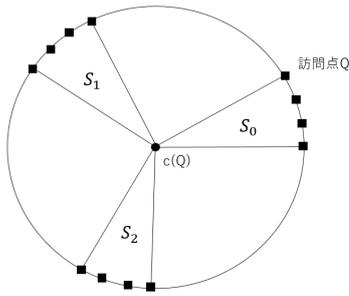


図 2 訪問点の集合が 3 個のセクターに分割される例. 訪問点を黒い四角で表す.

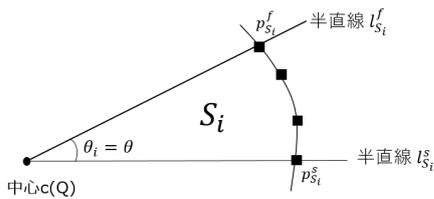


図 3 セクター  $S_i$

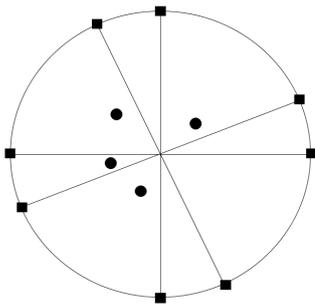


図 4 初期配置  $P$

$\rho(Q)$  の約数であり, すべての訪問点は訪問点の最小包含円上にあると仮定する. 初期配置ですべてのロボットは異なる地点にあり, 提案アルゴリズムの実行中も 2 台以上のロボットが重なることはない.

ロボットの台数  $n$  と訪問点の配置  $Q$  に対し, 警邏開始配置  $U(n, Q)$  とは,  $n$  台のロボットが正  $n$  角形の頂点上にあり, かつ, それぞれ  $c(Q)$  の円周とセクター右端の境界線  $l_{S_i}^s$  とが交差する点に位置していることである (図 4 に対する警邏開始配置を図 5 に示す).

提案手法では警邏方法として, 周回警邏とセクター警邏を考える. 隣り合った訪問点をそれぞれ結ぶことのできる,  $c(P)$  を囲むような閉路を, すべてのロボットが一巡して訪問する警邏方法を周回警邏と呼ぶ (図 6). 一方で, 各ロボットが 1 つあるいは複数のセクターに含まれる訪問点を担当し, 訪問する警邏方法をセクター警邏とする (図 7).

提案アルゴリズムは, 警邏法決定ステップ, 準備ステップ, 警邏ステップの 3 つのステップから構成される. 警邏法決定ステップでは, ロボットの台数  $n$  と訪問点の配置  $Q$

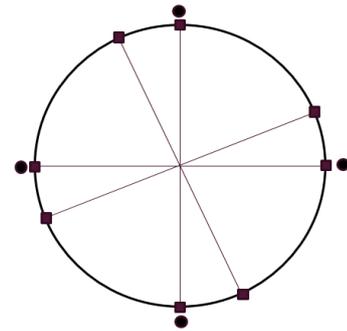


図 5 警邏開始配置  $U(n, Q)$ . ロボットを黒い点, 訪問点を黒い四角で表す.

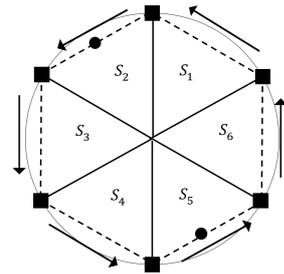


図 6 周回警邏の経路. ロボットを黒い点, 訪問点を黒い四角, 警邏経路を破線で表し, 矢印方向に移動する.

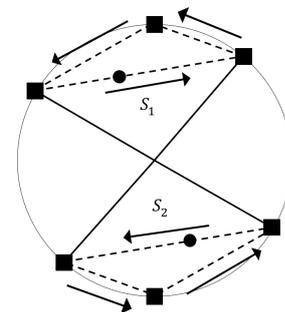


図 7 セクター警邏の経路. 記号は図 6 と同様.

をもとに, 各ロボットが周回警邏とセクター警邏から 1 つの警邏法を選ぶ. 準備ステップでは, 各ロボットが初期配置から警邏開始配置まで移動する. 警邏ステップでは, 各ロボットが求められた経路に沿って移動し, 訪問点を警邏する.

まず, 警邏法決定ステップにおいて,  $\rho(Q) = n$  かつセクターのなす角  $\theta < \pi/\rho(Q)$  のとき, 警邏法はセクター警邏とする. それ以外のとき, 警邏法は周回警邏とする.

準備ステップでは, ロボットの初期配置によって場合分けをする. 各ロボットはそれぞれの場合に応じて警邏開始配置まで移動する.

- $\rho(P) = 1$  のとき. 各ロボットの中心ビューはそれぞれ異なる. 各ロボットは, 中心ビュー最小のロボットをリーダーと決め, リーダーのロボットを基準として他のロボットとセクターの順序付けに合意できる. あと

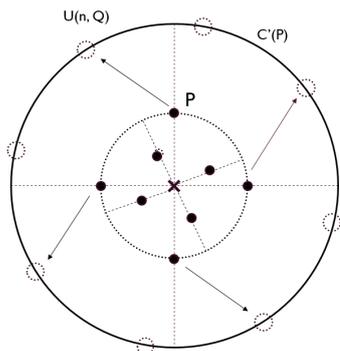


図 8  $1 < \rho(P) < n$  の場合, 最初に移動するロボット.

は対応するセクターの警邏開始配置に移動すればよい.

- $1 < \rho(P) < n$  のとき. 中心ビュー最小となる複数のロボットは, 点  $c(Q)$  を中心に時計回りで最も近い警邏開始配置の位置に移動する. その後, 移動しなかったロボットの中で, 中心ビュー最小となる複数のロボットが, まだ埋まっていない警邏開始配置を反時計回りで埋めるように移動する. これを繰り返す, すべてのロボットが警邏開始配置に移動する (図 8).
- $\rho(P) = n$  のとき. すでにロボットの配置は正  $n$  角形となっている. 各ロボットは正  $n$  角形を保ったまま, 点  $c(Q)$  を中心として反時計回りに最も近い警邏開始配置の位置にそれぞれ移動する.

警邏ステップでは, 各ロボットが計算した移動経路に沿って反時計回りに訪問点を警邏する.

$m = |Q|$  と提案アルゴリズムについて, 以下の定理が成立する.

**定理 1**  $m \neq 2n$  のとき, 上記のアルゴリズムで選択される警邏経路が最速である.

### 3.3 $m = 2n$ の場合の無記憶ロボットのための漸近的最速経路

**定理 2**  $m = 2n$  のとき, 最速の警邏方法は存在しない.

各ロボットがそれぞれ 2 個の訪問点を交互に訪問する場合を考える. このときの最短経路は, 対となる訪問点を結ぶことのできる線分に沿って, 各ロボットが往復するような移動経路である. ロボットは無記憶であるため, 経路上で停止した場合, 次のサイクルで自身が直前に訪問した訪問点がどちらであるかわからない. 経路は 1 本の線分であるため, 訪問点上でない地点で止まった場合, ロボットは次に向かうべき方向を決めることができない. したがって,  $m = 2n$  のとき, 最速の警邏方法は存在しない.

$m = 2n$  のときの経路は次のように設定することで漸近的に最速経路となる. ロボット  $r_i$  が警邏する 2 つの訪問点をそれぞれ点  $q_{i1}, q_{i2}$  とする. ここで  $Q$  の最小包含円の半径  $r$  よりも大きく, 2 点  $q_{i1}, q_{i2}$  を通るような半径  $r' (> r)$  の円  $C'$  を考える. このような円は 2 つあるが, 線分  $q_{i1}q_{i2}$

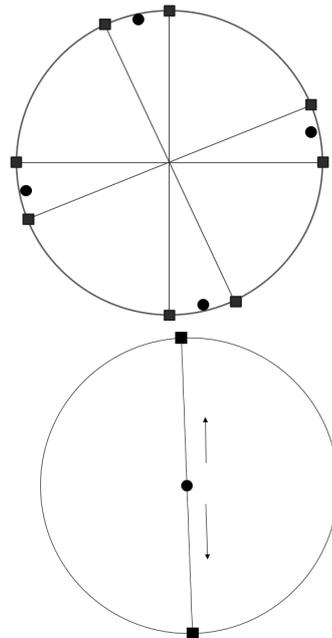


図 9 最速な警邏方法がない二つの例.

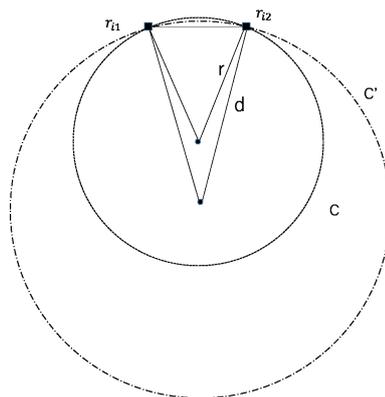


図 10 2 個の訪問点と新たな円  $C'$

に対し線対称なのでまず片方について考える. この円について, 線分  $q_{i1}, q_{i2}$  を両端とする弧のうち, 中心角の小さい弧を経路とする. もう一方の円についても同様にして, 弧を経路とする. こうして得られた経路はお互い交差せず, 2 点  $q_{i1}, q_{i2}$  を連結する単純閉路となる.

更に,  $r$  に比べて半径  $r'$  を極めて大きくすることで, 得られる経路は限りなく線分  $q_{i1}, q_{i2}$  に近づくことができる. つまり, 限りなく最短の経路に近い経路に沿って警邏できるので, 漸近的に最速の警邏をすることができる.

**定理 3** 上記の方法で定めた警邏は漸近的に最速の警邏をすることができる.

## 4. まとめと今後の課題

本稿では, ロボットの台数が訪問点の対称度の約数であり, かつ, 訪問点が最小包含円の円周上で存在する場合の分散警邏アルゴリズムを提案した.

今後の課題は, 任意の訪問点の配置に対する警邏アルゴ

リズム, ロボットの初期配置の対称度を利用した警邏アルゴリズムである.

#### 参考文献

- [1] Paola Flocchini, Giuseppe Prencipe, Nicola Santoro: Distributed Computing by Mobile Entities, Current Research in Moving and Computing. Springer 2019
- [2] Ichiro Suzuki, Masafumi Yamashita: Distributed Anonymous Mobile Robots: Formation of Geometric Patterns. SIAM J. Comput. 28(4): 1347–1363 (1999)
- [3] Masafumi Yamashita, Ichiro Suzuki: Characterizing Geometric Patterns Formable by Oblivious Anonymous Mobile Robots. Theor. Comput. Sci. 411(26–28): 2433–2453 (2010)
- [4] Nao Fujinaga, Yukiko Yamauchi, Hirofumi Ono, Shuji Kijima, Masafumi Yamashita: Pattern Formation by Oblivious Asynchronous Mobile Robots. SIAM J. Comput. 44(3): 740–785 (2015)
- [5] Shantanu Das, Paola Flocchini, Nicola Santoro, Masafumi Yamashita: Forming Sequences of Geometric Patterns with Oblivious Mobile Robots. Distrib. Comput. 28: 1347–1363 (2015)