

限られた視界を持つ1台のロボットによる線分被覆問題

門出 顕宏^{1,a)} 山内 由紀子² 来嶋 秀治² 山下 雅史²

概要：有限の視野 V を持つロボット群によって線分 L を被覆する問題を検討する。すなわち、 L 上のすべての点が、あるロボットからの距離が V 以下になるように、 L 上にロボット群を配置する問題である。ただし、各ロボットは L を離れて動くことはできない。この問題を難しくしているのは、(1) 各ロボットは同じアルゴリズムに従って動く、(2) 各ロボットは V を知らない、そして (3) 各ロボットは記憶を持たない（従って、アルゴリズムは、現在の視野内にあるロボット達の位置だけに従って、次に移動する点を決める）からである。まず、 $|L| \leq 2V$ を仮定して、1 台のロボットで L を被覆するアルゴリズムを提案し、その正当性を証明する。続いて、 n 台のロボットで L を被覆するアルゴリズムを提案するが、このアルゴリズムが正しく働くためにはパラメータを正しく設定する必要がある。

キーワード：線分被覆問題、自律型移動ロボット、無記憶

Covering a Line Segment by a Mobile Robot with Limited Visibility

Abstract: We investigate the problem of covering a given line segment L by a group of autonomous mobile robots with visibility radius V . That is, we investigate the problem of locate robots on L so that every point in L is within distance V from a robot. Note that the robots must stay on L anytime. What makes the problem difficult are (1) every robot obeys the same algorithm, (2) V is not available, and (3) each robot does not have a memory, so that the algorithm given to a robot needs to determine its next position from the current positions of the robots. Assuming $|L| \leq 2V$, we first propose an algorithm to cover L by a single robot and show its correctness. Next, we propose an algorithm to cover L by n robots. However, we need to appropriately select its parameter for the algorithm to work correctly.

Keywords: covering a line segment, autonomous mobile robot, oblivious

1. はじめに

本稿では、半径 V の視界を持つ1台の自律ロボットによって長さ L ($\leq 2V$) の線分を視界に収める問題を解くアルゴリズムをまず提案し、続いて、 n 台の自律ロボットによって長さ L の線分を被覆する問題を解くアルゴリズムを提案する。

2次元ユークリッド空間における、自律分散ロボットシステムに関する研究がこれまでに数多く行われている。自律分散ロボットシステムの研究では、システムを構成する各

ロボットが同じアルゴリズムに従い、自律的に動作することで、システム全体としてある目標を達成することを目指す。ロボットは2次元ユークリッド空間上の点としてモデル化する。主な問題として、ロボット全体で所望の幾何学的な配置を形成するパターン形成問題や、散らばっているロボットをある一点に集める一点集合問題がある [3], [4]。ロボットの能力に様々な制限を課し、それらの制限の下である問題を解くことができるかどうかを判定する問題も活発に研究されている。

本稿で検討する問題は、ある線分が与えられ、その線分上の任意の点に各ロボットが重複しないように配置されているとき、線分上の各点があるロボットの視界で被覆できるように各ロボットがその位置を自律的に変えるアルゴリズムを構成する問題である。ただし、各ロボットを節点とし、互いに視界に入るロボットの対を辺で結んだ無向グ

¹ 九州大学電気情報工学科
Department of Electrical Engineering and Computer Science, Kyushu University

² 九州大学大学院システム情報科学研究院
Graduate School of Information Science and Electrical Engineering, Kyushu University

^{a)} monde@tcslab.csce.kyushu-u.ac.jp

ラフはアルゴリズムの終了時点において連結でなければならない。この問題を解くにあたり、ロボットはいくつかの協調的な動作を行わなければならない。第1に、ロボットは最悪時には等間隔に配置される必要がある (例えば, $L = (n+1)V$ のとき)。第2に、ロボットの移動経路が交差してはならない。現実の質量を持つ物理的なロボットに対してアルゴリズムを適用したとき、衝突が起きては困るからである。ロボットの視界を無限と仮定して、線分上にロボットを等間隔に配置するアルゴリズムが [1] で提案されている。また、侵入者を感知するセンサーの半径が r で、他のロボットを確認できる視界の半径を $2r$ としたとき、各ロボットが持つセンサーで線分を被覆するアルゴリズムが [2] で提案されている。文献 [2] では、ロボットが r を知っているとは仮定しているが、本研究ではロボットの持つ視界半径 V を有限とし、ロボットは V を知ることができないと仮定する。

ロボットの総数を n , 視界半径を V , 線分の長さを L とする。この問題を難しくしているのは、(1) 各ロボットは同じアルゴリズムに従って動く、(2) 各ロボットは V を知らない、そして (3) 各ロボットは記憶を持たない (従って、アルゴリズムは、現在の視野内にあるロボット達の位置だけに従って、次に移動する点を決める) からである。事実、図1に示すように、すべてのロボットを等間隔 V で配置しなければならないときに、ロボットが周囲の状況から V , すなわち, $\frac{L}{n+1}$ を計算することは困難である。そこで、問題を簡単にするために、まず $L \leq 2V$ を仮定し、1台のロボットで L を被覆するアルゴリズム SCOR を提案する。ロボットが1台の場合にも、上述の (2) と (3) の条件があり、問題は十分に難しい。最も難しいのは $L = 2V$ のときで、この場合、ロボットは線分の中点へ移動しなければならない。仮に、(2) の制約を外すと、ロボットは見えている端点から V だけ離れれば両端点が必ず視界に入るため、この問題を簡単に解決する。また、(3) の制約を外せば、ロボットは過去に自分が進んだ距離 (図2における x) を記憶できるため L を計算でき、線分の中点へ移動することができる。直感に反して、 V を知らず、記憶を持たないロボットが V を計算することは困難を極める。SCOR は拡大体の性質を用いて、図2の y から L を計算する。本稿では、 $V \geq 1$ かつ $L \in \mathbb{Q}$ ($\mathbb{Q}$: 有理数全体) という条件の下で $L = 2V$ のとき、SCOR の正当性を示す。SCOR は、“ y を有理数体 $\mathbb{Q}$ の有限次拡大体とすることで、 y の有理数部分を L とする” というアイデアに基づいているので、 $L \in \mathbb{Q}$ は重要な条件である。実際に、 $L \in \mathbb{R}$ の場合に直接拡張することは困難と思われる。

SCOR を n 台のロボットに適用できるように拡張することも難しい。そこで、 $n > (1 + \frac{1}{m-1})\frac{L}{V} - 1$ を仮定した上で、 n 台のロボットで L を被覆するアルゴリズム SCA を提案する。ここで、 m はアルゴリズムに与えるパラメータ

であるが、これが正しく設定されていない場合、SCA の正当性は保証されないという意味で、SCA は発見的アルゴリズムである。

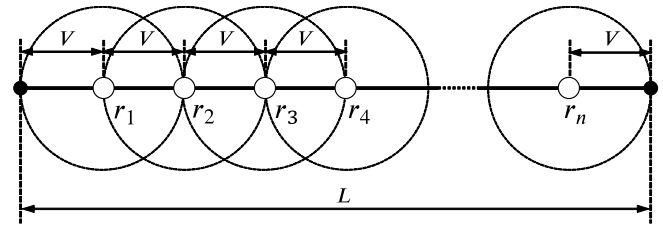


図1 $L = (n+1)V$ の場合において、問題を解決するロボットの配置。

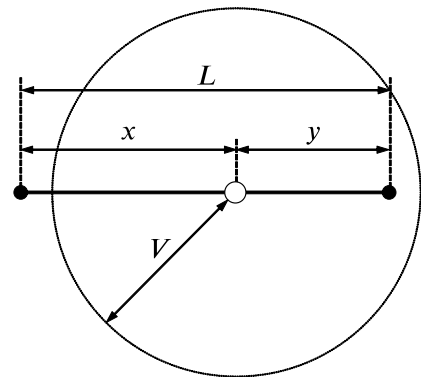


図2 $L = 2V$ の場合の1台のロボットの配置。

2. 準備

2.1 ロボットモデル

本論文のロボットシステムのモデルは、[3] で提案されたモデルに基づいている。2次元平面上に存在する n 台のロボットの集合を $R = \{r_1, r_2, \dots, r_n\}$ とする。すべてのロボットは平面上の点とする。よって、複数のロボットが同一点に存在することや、ロボットが移動する際他のロボットと交差することが起こり得る。この2次元平面上に大域座標系を $x-y$ 直交座標系で定義する。大域座標系における、 r_i の時刻 t での座標を $p_i(t)$ とし、集合 $P(t) = \{p_1(t), p_2(t), \dots, p_n(t)\}$ を時刻 t における R の配置という。 $t = 0$ における配置 $P(0)$ を初期配置と呼び、本システムにおいては離散の時刻 $t = 0, 1, 2, \dots$ について考える。各ロボットは、大域座標系にはアクセスできず、それぞれが自身を原点とする $x-y$ 直交座標系を局所座標系として持つ。 r_i が持つ局所座標系を Z_i とし、 $z_i(p)$ を Z_i における点 p の座標とする。ゆえに、任意の時刻 t で $z_i(p_i(t)) = (0, 0)$ である。図3に示すように、各 Z_i について、単位距離、 x 軸方向、及び右手系であるか左手系であるかの合意はない。また、すべてのロボットは、お互いを識別できず、匿名性をもっており、ロボット間での通信はできない。そのため、ロボットのすべての動作は、自身の

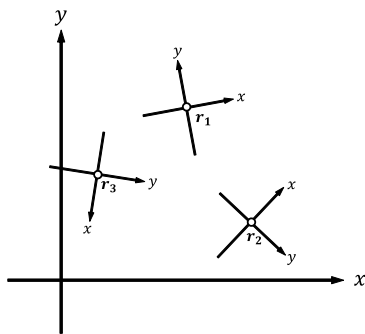


図 3 大域座標系と局所座標系.

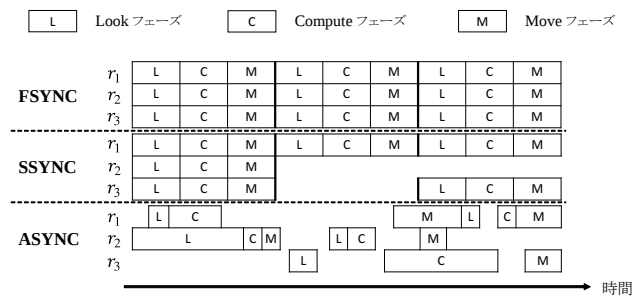


図 4 各同期モデルの例.

局所座標系にのみ基づき、全体として自律したシステムになっている。さらに、各ロボットに与えるアルゴリズムはすべて同じものである。

各ロボットの一連の動作は、3つのフェーズ Look-Compute-Move のサイクルの繰り返しからなる。上述のように、これらのフェーズにおけるロボットの動作は、自身の局所座標系に基づいている。Look フェーズでは、視界内の周囲の状態を観測する。Compute フェーズでは、Look フェーズで観測した結果を入力として、与えられたアルゴリズムに従い次の移動先を計算する。Move フェーズでは、Compute フェーズで得られた移動先へ移動する。先行研究において、このサイクル及び、フェーズの実行されるタイミングによって、一般に以下の三つの同期モデル [3] に分けられている。任意の $i \geq 1$ について、すべてのロボットが i 回目の Look-Compute-Move サイクル及び、各フェーズを同期して必ず実行するモデルを完全同期モデル (**FSYNC**) という。ある i について、いくつかのロボットが i 回目の Look-Compute-Move サイクルを実行しないことがあるが、 i 回目のサイクルを実行しているロボットについては、サイクル及び、各フェーズを同期して実行するモデルを準同期モデル (**SSYNC**) という。サイクル及びフェーズの実行に仮定を置かないモデルを非同期モデル (**ASYNC**) という。これらのモデルを図式的に表したものを図 4 に示す。図 4 における空白は、ロボットがどのフェーズも実行していない状態を表す。本研究では、**FSYNC** モデルを仮定する。すなわち、すべてのロボットは各時刻において同期的に必ず Look-Compute-Move サイクルを実行する。

すべてのロボットは、実行履歴を記録するためのメモリを有しておらず、このようなロボットを無記憶ロボット [3] という。そのため、ロボットに与えるアルゴリズムにそれ以前の状態に関する情報は使用できず、アルゴリズムは直前の Look フェーズにより得られた情報のみから、次の移動先を計算しなければならない。

また、各ロボットは半径 V の円盤状の限られた視界を持つ。 r_i の持つ視界内のすべての点の集合を V_i とする。ロボットは、視界内に存在する、他のロボットと 2 次元平面上に与えられた線分の視界内の部分、及び、端点を認識できるものとする。これにより、ロボットは与えられた線分上を移動することが可能となる。

2 次元平面上に与えられる線分を s とする。この線分 s の端点の集合を $F = \{f_1, f_2\}$ とする。また、線分 s の長さを L とする。 Z_i における視界内に存在する s 上の点の集合を $Z_i(s)$ とする。さらに、 $Z_i(P(t) \cup F) = \{z_i(p) | p \in (P(t) \cup F) \cap V_i\}$ とする。ロボットは、与えられたあるアルゴリズム A に従い移動先を計算するが、アルゴリズムにはロボットの総数 n 、各ロボットの持つ視界半径 V 、及び、与えられた線分長さ L の値を与えないものとする。以上で述べたロボットモデルにおいて、アルゴリズム A は、 $Z_i(P(t) \cup F)$ と $Z_i(s)$ を入力とし、次のロボットの移動先 $Z_i(p)$ ($p \in \mathbb{R}^2$) を出力とする関数で表される。

2.2 隣接可視グラフ

線分の端点の名前を e_1, e_2 とする。新たな集合 $R' = R \cup \{e_1, e_2\}$ を定義する。時刻 t において、 r_i と r_j ($r_i, r_j \in R$) が互いに隣のロボットであると認識できるとき、かつそのときに限り、辺 $(r_i, r_j) \in E_v(t)$ を定義する。ここで、ロボットの隣接可視グラフを無向グラフ $G_v(t) = (R', E_v(t))$ で定義する。辺 $(r_i, r_j) \in E_v(t)$ が存在するとき、時刻 t においてロボット r_i, r_j が連結であるといい、隣接可視グラフ $G_v(t)$ が連結であるとき、時刻 t においてすべてのロボットが連結であるという。また、各時刻 t において、 $V(r_i) = \{r | (r_i, r) \in E_v(t)\}$ を定義する。すなわち、 $|V(r_i)| = 0, 1$ または 2 である。

2.3 線分被覆問題

与えられたある線分上のすべての点を、各ロボットの持つ視界で被覆する問題を以下では検討する。線分被覆問題の明確な定義を以下では行う。

次のように各記号を定義する。

- ロボットの集合を $R = \{r_1, r_2, \dots, r_n\}$ とする。
- 時刻 t での R の配置を $P(t)$ とする。
- ロボット $r_i \in R$ の持つ視界内のすべての点の集合を V_i とする。

- すべてのロボットが連結であり、かつ、

$$s \subset \bigcup_{1 \leq i \leq n} V_i$$

となる n 台のロボットの配置の集合を C とする。

定義 1 線分被覆問題のインスタンス I は、3 項組 (s, n, V) である。ここで、 s は線分、 n はロボットの総数、 V はロボットの視界の半径である。 I が与えられたとき、任意の初期状態 $P(0)$ から出発する実行列 $\varepsilon: P(0), P(1), \dots$ が C に属する状態 $P(t)$ を含むとき I を解くという。

3. アルゴリズム SCOR

本章では、1 台のロボットによって、線分被覆問題を解くアルゴリズム SCOR (Segment-Cover-for-One-Robot) を提案する。SCOR は、ロボットの視界半径が $V \geq 1$ 、線分の長さが $L \in \mathbb{Q}$ であるとき、 $L = 2V$ を仮定して、1 台のロボットで長さ L の線分を被覆するアルゴリズムである。すなわち、ロボットを線分の中点に移動させるアルゴリズムである。

F を体 E の部分体、 T を体 E の部分集合としたとき、 F 上 T で生成される体を $F(T)$ で表す ([5], p48 参照)。 R を単位元を持つ多項式環とする。このとき、 R の元を係数とする n 変数 $X_1, \dots, X_n$ の多項式環を $R[X_1, \dots, X_n]$ で表す ([5], p28 参照)。 [5] によれば、次の補題が成り立つ。

補題 2 (定理 2.1[5]) F を体 E の部分体、 T, T_1, T_2 を E の部分集合とすると、

- (1) $F(T_1 \cup T_2) = F(T_1)(T_2)$
- (2) T の有限部分集合を一般に S で表すとき、 $F(T) = \bigcup_S F(S)$ である。
- (3) $T = \{t_1, \dots, t_n\}$ が有限集合のとき、

$$F(T) = \left\{ \frac{f(t_1, \dots, t_n)}{g(t_1, \dots, t_n)} \mid f, g \in F[X_1, \dots, X_n], g(t_1, \dots, t_n) \neq 0 \right\}.$$

$\mathbb{Q}$ を有理数全体を表す集合とする。補題 2 より、以下の定理が導かれる。

定理 3 $p_1, \dots, p_i$ を互いに相異なる素数とし、 p_j はこのうちのどの素数とも異なる素数とする。このとき、 $\frac{1}{\sqrt{p_j}} \notin \mathbb{Q}(p_1, \dots, p_i)$ 。

$\mathbb{P}$ を素数全体を表す集合とし、 $\pi_i \in \mathbb{P}$ を小さい方から i 番目の素数とする。また、任意の自然数 i について、 $\mathbb{Q}_i = \{\alpha_0 + \alpha_1 \frac{1}{\sqrt{\pi_1}} + \dots + \alpha_i \frac{1}{\sqrt{\pi_i}} \mid \alpha_j \in \mathbb{Q}, \pi_j \in \mathbb{P}\}$ とする。

系 4 $j \geq i + 1$ である、任意の i, j について、 $\alpha_j \neq 0$ 、 $\alpha_k \in \mathbb{Q}$ ($0 \leq k \leq j$) のとき、 $\alpha_0 + \alpha_1 \frac{1}{\sqrt{\pi_1}} + \dots + \alpha_k \frac{1}{\sqrt{\pi_k}} + \dots + \alpha_j \frac{1}{\sqrt{\pi_j}} \notin \mathbb{Q}_i$ 。

証明. 定理 3 より、 $\alpha_0 + \alpha_1 \frac{1}{\sqrt{\pi_1}} + \dots + \alpha_k \frac{1}{\sqrt{\pi_k}} + \dots + \alpha_j \frac{1}{\sqrt{\pi_j}} \notin \mathbb{Q}(\pi_1, \dots, \pi_i)$ 。また、 $\mathbb{Q}_i \subset \mathbb{Q}(\pi_1, \dots, \pi_i)$ であるから、 $\alpha_0 + \alpha_1 \frac{1}{\sqrt{\pi_1}} + \dots + \alpha_k \frac{1}{\sqrt{\pi_k}} + \dots + \alpha_j \frac{1}{\sqrt{\pi_j}} \notin \mathbb{Q}_i$ 。□

系 4 を用いて、SCOR は、線分被覆問題を解決する。図 5 に時刻 $t \geq 0$ におけるロボットの配置を示す。時刻 t における、ロボットと左端点との距離を $x(t)$ 、ロボットと右端点との距離を $y(t)$ とする。集合 T_i を、 $T_i = \{\alpha - (\frac{1}{\sqrt{\pi_1}} + \frac{1}{\sqrt{\pi_2}} + \dots + \frac{1}{\sqrt{\pi_i}}) \mid \alpha \in \mathbb{Q}\}$ とする。また、 $\beta_i = \{\frac{1}{\sqrt{\pi_1}} + \frac{1}{\sqrt{\pi_2}} + \dots + \frac{1}{\sqrt{\pi_i}}\}$ とする。以下に 1 台のロボット r_1 に対する、線分被覆問題を解くアルゴリズム SCOR を示す。

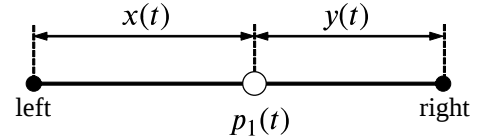


図 5 時刻 t におけるロボットの配置。

Algorithm 1 $r_1 \in R$ に対するアルゴリズム SCOR

```

1: if 左端点が見える then
2:   if 右端点が見える then
3:     停止する。
4:   else
5:     if 任意の  $i$  について、 $x \neq \beta_i$  then
6:       端点から  $\frac{1}{\sqrt{\pi_1}}$  だけ離れた位置に移動。
7:     else
8:        $x = \beta_i$  となる  $i$  について、 $x + \frac{1}{\sqrt{\pi_{i+1}}}$  だけ端点から
       離れた位置に移動。
9:   else
10:    if 右端点が見える then
11:      if 任意の  $i$  について、 $y \notin T_i$  then
12:        端点から  $2y$  だけ離れた位置に移動。
13:      else
14:        if  $y \in T_i$  となる  $i$  について、 $\frac{1}{2}(y + \sum_{k=1}^i \frac{1}{\sqrt{\pi_k}}) > 2y$ 
        then
15:          端点から  $2y$  だけ離れた位置に移動。
16:        else
17:           $\frac{1}{2}(y + \sum_{k=1}^i \frac{1}{\sqrt{\pi_k}})$  だけ端点から離れた位置に
          移動。
18:      else
19:        停止する。

```

4. SCOR の正当性

ある時刻 t におけるロボットの状態を次の 8 つに場合分けする。 $y(t) \in T_i$ のとき、 $y'(t) = \frac{1}{2}(y(t) + \sum_{k=1}^i \frac{1}{\sqrt{\pi_k}})$ とする。

1. 左端点が見えている。(状態 1)
2. 右端点が見えている。
- 2.1. 任意の i について、 $y(t) \notin T_i$ 。

- 2.1.1. $y(t) \notin \mathbb{Q}$. (状態 2)
 2.1.2. $y(t) \in \mathbb{Q}$. (状態 3)
 2.2. ある i について, $y \in T_i$.
 2.2.1. $y'(t) > 2y(t)$. (状態 4)
 2.2.2. $y'(t) \leq 2y(t)$.
 • $y'(t) > \frac{L}{2}$. (状態 5)
 • $y'(t) < \frac{L}{2}$. (状態 6)
 • $y'(t) = \frac{L}{2}$. (状態 7)
 3. 両端点が見えている. (状態 8)

状態 8 は問題を解決している状態である. 以下では, ロボットの視界半径が $V \geq 1$, 線分の長さが $L \in \mathbb{Q}$, $L = 2V$ であると仮定する. いくつかの補題を示し, ロボットの初期配置での状態が, 状態 1–7 のいずれであっても有限時間内に状態 8 になることを示す.

補題 5 (定理 1.4[6])

$$\sum_p \frac{1}{p}$$

は発散する. ここに和はすべての素数 p について渡る.

補題 5 より, 以下の補題が導かれる.

補題 6 素数の平方根の逆数の和は, 無限大に発散する. すなわち,

$$\sum_{i=1}^{\infty} \frac{1}{\sqrt{\pi_i}} = \infty.$$

補題 6 より, 左端点のみが見えている状態から右端点が見える状態へ必ず遷移するため, 補題 7 が成り立つ. さらに以下の補題が成り立つ.

補題 7 状態 1 から有限時間内に状態 7 へ遷移する.

補題 8 状態 2 から有限時間内に状態 1, 4, 5, 6 あるいは 7 のいずれかへ遷移する.

補題 9 状態 3 から有限時間内に状態 1, 8 のどちらかへ遷移する.

補題 10 状態 4 から 1 単位時間で状態 2 へ遷移する. さらに, 一度状態 4 から状態 2 へ遷移した後, 状態 4, 5, 6 あるいは 7 へ遷移することはない.

補題 11 状態 5 から 1 単位時間で状態 1 へ遷移する.

補題 12 状態 6 から 1 単位時間で状態 3 へ遷移する.

補題 13 状態 7 から 1 単位時間で状態 8 へ遷移する.

補題 10 より, 一度状態 4 から状態 2 へ遷移した後, 状態 4, 5, 6 あるいは 7 となることはない. 状態 4, 5, 6 及び 7 に遷移しないような新たな状態 2' を作る. 状態 2' からは状態 1 にのみ遷移する.

定理 14 $V \geq 1$ かつ $L \in \mathbb{Q}$ であると仮定すると, SCOR は $L = 2V$ のとき 1 台のロボットで線分被覆問題を解く.

証明. $V \geq 1$, $L \in \mathbb{Q}$, $L = 2V$ の仮定の下, 成り立つ補題 7–13 より, 新たな状態 2' を加えた全体の状態遷移図を図 6 に示す. すべての遷移について, 遷移は有限時間内に起こる. 図 6 より, 初期配置における状態がどの状態であっても, 終端配置では状態 8 となる. すなわち, 問題を解決するので, 定理 14 が証明された. $\square$

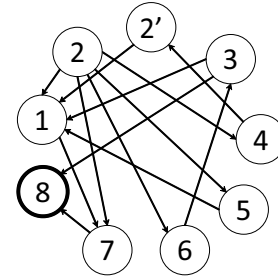


図 6 状態遷移図

5. アルゴリズム SCA

本章では, n 台のロボットからなる自律分散ロボット群による線分被覆問題を解くアルゴリズムを示す. $n \geq \frac{L}{V} - 1$ を満たせば, 線分全体を視界で被覆するのに十分である. 提案アルゴリズム SCA (Segment-Cover-Algorithm) は, $n > (1 + \frac{1}{m-1})\frac{L}{V} - 1$ を仮定して線分被覆問題を解決する. ただし, $m > 1$ はアルゴリズムに与えるパラメータである. SCA では, ロボットどうしの衝突や交差を起こさない. $dist(p_i, p_j)$ をロボット r_i, r_j 間のユークリッド距離と定義する.

Algorithm 2 $r_i \in R$ に対するアルゴリズム SCA(m)

```

1: if  $V(r_i) = \{r_j\}$  then
2:    $r_i$  は  $r_j$  から遠ざかる方向に  $\frac{1}{2m} dist(p_i, p_j)$  だけ移動.
3: else
4:   if  $V(r_i) = \{r_{i-1}, r_{i+1}\}$  then
5:     if  $dist(r_i, r_{i-1}) > dist(r_i, r_{i+1})$  then
6:        $r_i$  は  $r_{i+1}$  から遠ざかる方向に,
        $\frac{dist(p_i, p_{i-1}) - dist(p_i, p_{i+1})}{2}$  だけ移動.
7:     else
8:        $r_i$  は  $r_{i-1}$  から遠ざかる方向に,
        $\frac{dist(p_i, p_{i-1}) - dist(p_i, p_{i+1})}{2}$  だけ移動.
9:   else
10:    停止する.
```

6. SCA の正当性

本章では、 $n > (1 + \frac{1}{m-1})\frac{L}{V} - 1$ を満たすとき、SCA は線分被覆問題を解決することを示す。

隣接するロボットを結ぶ辺を $q_i = (r_i, r_{i+1})$ とする。また、両端のロボットと線分の端点について、 $q_0 = (e_1, r_0), q_n = (r_n, e_2)$ とし、 $Q = \{q_i | i = 0, 1, \dots, n\}$ とする。時刻 t における各ロボットの位置を $p_i(t)$ 、 r_i と r_{i+1} の間の距離を $d_i(t) = \text{dist}(p_i(t), p_{i+1}(t))$ とし、 $D(t) = \{d_i(t) | i = 0, 1, \dots, n\}$ とする (図 7)。 $D(t)$ の要素のうち、最小値を $d_{\min}(t)$ とする。時刻 t において、SCA をロボット群に適用したときの各ロボットの移動先は、次の 4 つの場合に分けられる。

- (1) 両隣のロボット r_{i-1} と r_{i+1} が見える場合：

$$p_i(t+1) = p_i(t) + \frac{d_i(t) - d_{i-1}(t)}{2}.$$

- (2) 隣のロボット r_{i-1} のみが見える場合：

$$p_i(t+1) = p_i(t) + \frac{d_{i-1}(t)}{2m}.$$

- (3) 隣のロボット r_{i+1} のみが見える場合：

$$p_i(t+1) = p_i(t) + \frac{d_{i+1}(t)}{2m}.$$

- (4) 他のロボットが見えない場合：

$$p_i(t+1) = p_i(t).$$

$d_i(t+1)$ は、 r_i と r_{i+1} が上の 4 つの場合のいずれであるかによって決まる。可能な組み合わせ $h_i(t) = (r_i \text{ の場合}, r_{i+1} \text{ の場合})$ は、 $(1, 1), (1, 2), (2, 3), (2, 4), (3, 1), (3, 2), (4, 3)$ 及び $(4, 4)$ である。

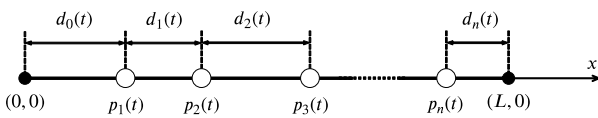


図 7 時刻 t における、ロボット間距離 $d_i(t)$ 。

補題 15 $n > (1 + \frac{1}{m-1})\frac{L}{V} - 1$ を仮定する。任意の時刻 $t \geq 0$ について、 $d_{\min}(t+1) \geq d_{\min}(t)$ が成り立つ。

任意に $t_0 \geq 0$ を固定する。 $t \geq t_0$ に対して、 $M^{t_0}(t) = \{q_i | d_i(t) = d_{\min}(t)\}$, $H_1^{t_0}(t) = \{(q_{i-1}, q_{i+1}) | q_{i-1}, q_{i+1} \in M^{t_0}(t), 1 \leq i \leq n-1\}$, $H_2^{t_0}(t) = \{(q_0, q_1) | q_0, q_1 \in M^{t_0}(t)\}$, $H_3^{t_0}(t) = \{(q_{n-1}, q_n) | q_{n-1}, q_n \in M^{t_0}(t)\}$ と定義する。さらに、 $H^{t_0}(t) = H_1^{t_0}(t) \cup H_2^{t_0}(t) \cup H_3^{t_0}(t)$ とする。

$|M^{t_0}(t)| = n+1$ であれば線分被覆問題は解決されている。解決していないとすると、 $|M^{t_0}(t)| \leq n$ である。

補題 16 $|M^{t_0}(t)| \leq n$ のとき、 $|H^{t_0}(t)| \leq |M^{t_0}(t)| - 1$ 。

補題 15 と補題 16 より、次の補題 17 が成り立つ。

補題 17 任意の時刻 t について、 $d_{\min}(t) < \frac{L}{n+1}$ ならば、 $d_{\min}(t)$ は広義の単調増加関数である。特に、任意の t について、 $d_{\min}(t) < d_{\min}(t+n)$ である。

定理 18 $n > (1 + \frac{1}{m-1})\frac{L}{V} - 1$ を満たすとき、SCA は線分被覆問題を解く。

証明。 線分被覆問題を解決していない任意の時刻 t について、 $d_{\min}(t) < \frac{L}{n+1}$ であるので、補題 17 より、 $d_{\min}(t)$ は広義の単調増加関数である。特に、任意の t について、 $d_{\min}(t) < d_{\min}(t+n)$ であるから、 n 時刻の間にロボット間距離の最小値は必ず増加する。よって、最小値は $\frac{L}{n+1}$ に収束するので、与えられた線分上の各点はあるロボットの視界で被覆される。すなわち、SCA は線分被覆問題を解く。□

7. おわりに

ロボットが 1 台の場合に $L \in \mathbb{Q}$ と $V \geq 1$ を仮定して、 $L = 2V$ のときに線分被覆問題を解決するアルゴリズム SCOR を提案した。また、 $n > (1 + \frac{1}{m-1})\frac{L}{V} - 1$ を仮定して、自律ロボット群による線分被覆問題を解くアルゴリズム SCA を提案したが、SCA は $L = (n+1)V$ 、あるいは適切な m が与えられなかった場合には線分被覆問題を解決できず、この意味で発見的アルゴリズムに留まる。今後は、残されている $L = 2V \in \mathbb{R}$ の場合に、1 台のロボットによる線分被覆アルゴリズムを研究する予定である。

参考文献

- [1] R. Cohen and D. Peleg, “Local spreading algorithms for autonomous robot systems,” Theoretical Computer Science, **399**, pp.71–82, 2008.
- [2] M. Eftekhari, P. Flocchini, L. Narayanan, J. Opatrny, and N. Santoro, “Distributed Barrier Coverage with Re-locatable Sensors,” Proc. of the 21st International Colloquium on Structural Information and Communication Complexity, pp.235–249, 2014.
- [3] N. Fjinaga, Y. Yamauchi, H. Ono, S. Kijima, and M. Yamashita “Pattern Formation by Oblivious Asynchronous Mobile Robots,” SIAM J. Comput., **44.3** (2015), pp.740–785.
- [4] H. Ando, Y. Oasa, I. Suzuki, and M. Yamashita “Distributed memoryless point convergence algorithm for mobile robots,” IEEE Trans. Robotics Automation, **15** (1999), pp.818–828.
- [5] 藤崎源二郎, 体とガロア理論, 岩波書店, 1991.
- [6] W. ナルキエヴィッチ, 素数定理の進展 (上), シュプリンガー・ジャパン, 2008.