

自律モジュールロボットによる二次元格子領域の探索

土居 圭介^{1,a)} 山内 由紀子^{2,b)} 来嶋 秀治^{2,c)} 山下 雅史^{2,d)}

概要：本研究では、格子状に区切られた $m \times n$ の長方形の二次元領域内に置かれた目標物をモジュールロボットで探索する探索問題を扱う。モジュールロボットは複数の同一なモジュールで構成されており、各モジュールは自律的に動作する。各モジュールは匿名で、無記憶（メモリを持たない）であり、また、格子領域全体を観測できず、目標物の位置を事前には知らない。さらに、モジュール間は明示的な通信を行わず、各モジュールが観測範囲の他のモジュールの配置をもとに移動することでモジュールロボット全体としての協調動作を実現する。モジュールロボットは任意の位置から探索を開始し、目標物の存在するセルにいずれかのモジュールが入ることで目標物を発見する。本研究では各モジュールに与える探索アルゴリズムを示す。

キーワード：分散アルゴリズム、モジュールロボット、探索問題。

Exploration by a Modular Robot in a Two-Dimensional Grid Field

KEISUKE DOI^{1,a)} YUKIKO YAMAUCHI^{2,b)} SHUJI KIJIMA^{2,c)} MASAFUMI YAMASHITA^{2,d)}

Abstract: We consider the **exploration problem** by a **modular robot (or metamorphic robot)** in a two-dimensional $m \times n$ grid field. The goal is to find a target placed in an arbitrary cell in the field. A modular robot consists of multiple uniform **modules** each of which autonomously moves in the field. Each module is anonymous, oblivious (memory-less), cannot observe the entire field, and does not know the position of the target beforehand. Additionally, the modules do not have explicit communication medium so that they cooperate with each other by just observing the positions of other modules in the visibility range. A modular robot starts exploration from an arbitrary initial configuration and finds the target by some module entering the cell where the target is placed. We present distributed exploration algorithms for each module.

Keywords: Distributed algorithm, modular robot, exploration problem.

1. はじめに

モジュールロボットとは、二次元格子領域中に置かれた複数の同一のモジュールから構成されるロボットである。単位格子に囲まれた領域をセルと呼び、各モジュールは1つのセル内に置かれている。個々のモジュールがスライド、

回転と呼ばれる単純な動きを行うことにより、モジュールロボット全体が移動する。本研究では、 $m \times n$ 二次元格子領域に置かれた目標物をモジュールロボットが探し出す探索問題を考える。

モジュールロボットは分子ロボットの計算モデルとして有用であると考えられており、走性、特に移動速度について研究が行われている。Dumitrescu ら [1] は、各モジュールがモジュールロボット全体の配置を知っており、方角（東西南北）について認識しているモジュールロボットの水平（東西）方向への移動速度が高々 $1/2$ であることを示し、モジュールロボットが予め決められた初期配置からの移動速度を実現する分散アルゴリズムを提案した。

¹ 九州大学工学部

School of Engineering, Kyushu University

² 九州大学大学院システム情報科学研究院

Faculty of Information Science and Electrical Engineering, Kyushu University

a) doi@tcslab.csce.kyushu-u.ac.jp

b) yamauchi@inf.kyushu-u.ac.jp

c) kijima@inf.kyushu-u.ac.jp

d) mak@inf.kyushu-u.ac.jp

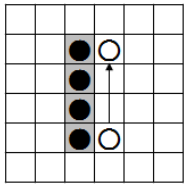


図 1 スライド移動

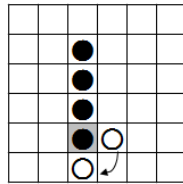


図 2 回転移動

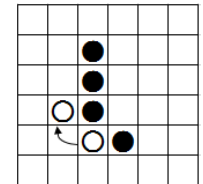
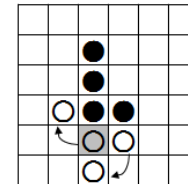
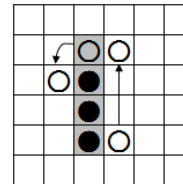


図 3 禁止される移動の例

Chen ら [2] は、各モジュールの視界が 3 近傍であり、方角について認識しているモジュールロボットが水平に一直線の配置になるまで移動し、 $1/3$ の速度で移動することを実現するアルゴリズムを提案した。また、各モジュールの視界が 4 近傍であり、方角について認識しているモジュールロボットが三次元格子領域上でモジュールロボットが x 軸の正方向へ $1/2$ の速度で移動する分散アルゴリズムを提案した。

本研究では、モジュールロボットが領域内に置かれた目標物を発見する探索問題を解くための分散アルゴリズムを提案する。ロボットシステムが目標物を見つけるためには、目標物の置かれたセルにいずれかのモジュールが入らなければならないものとする。本研究では、格子領域の方角（東西南北）を認識しているモジュールロボット、認識していないモジュールロボットのそれぞれについて探索問題を解く分散アルゴリズムを示す。

2. 準備

本章ではモジュールロボットのモデルを定義する。本稿では、 $m \times n$ の格子領域の上に配置されたモジュールを考える。横軸を x 軸、縦軸を y 軸とする。さらに、 x 軸の正方向を東、負方向を西とし、 y 軸の正方向を北、負方向を南とする。単位格子に囲まれた領域をセルと呼ぶ。 (x, y) , $(x, y+1)$, $(x+1, y)$, $(x+1, y+1)$ で囲まれたセルをセル (x, y) と呼ぶ。領域外である $(x < 0) \vee (m \leq x) \vee (y < 0) \vee (n \leq y)$ のセル (x, y) を壁と呼び、以降の図では黒塗りのセルとして表す。

ここでセル (x, y) は周囲に 8 つのセルを持つ。東のセルは $(x+1, y)$ 、南東のセルは $(x+1, y-1)$ 、南のセルは $(x, y-1)$ 、南西のセルは $(x-1, y-1)$ 、西のセルは $(x-1, y)$ 、北西のセルは $(x-1, y+1)$ 、北のセルは $(x, y+1)$ 、北東のセルは $(x+1, y+1)$ であり、東 (E)、西 (W)、南 (S)、北 (N) のセルはセル (x, y) に隣接すると言う。本稿ではモジュールが東西南北を認識している場合とそうでない場合を考える。

各モジュールは無記憶であり、視界の範囲で得られる情報のみから次の移動を決定する。各モジュールは匿名であり、同一のアルゴリズムに従って同期的に移動を行う。モジュールはセルの中しか置けず、1 つのセルには 1 つのモジュールしか置くことができない。また、各モジュールは今自身が存在しているセルの座標を知ることはできない。

各モジュールは、決まった範囲の視界を持ち、その視界の範囲内の各セルに、モジュールが存在するか、モジュールが存在しないか、壁であるかの 3 通りを識別できる。視界内に目標物が含まれていても識別することはできない。

本稿では全モジュールが各時刻に、視界内のセルの状態を識別する観測と観測結果をもとに移動を行う完全同期モデルを考える。

モジュールの移動方法としてはスライドと回転の 2 つの方法がある。

- スライド：あるモジュールが、南北あるいは東西に一直線に隣接しているモジュールを伝って 1 方向にいくつかのセルを移動する方法である。モジュールがスライド移動するとき、通過するセルの片側全てに、このモジュールと同時に移動しないモジュールが存在するとする。これらの移動しないモジュールを軸と言う（図 1）。
- 回転：あるモジュールが、隣接している灰色のセル内の 1 つを軸として、時計回りあるいは反時計回りに 90 度回転する移動方法である（図 2）。軸となるモジュールは回転するモジュールと同時に動いてはならない。

図 3 に移動の軸となっているモジュールが同時に移動できない例を示す。

次に、モジュールの連結性について述べる。モジュールの連結性を定義するために、モジュールロボットが構成するモジュールグラフを考える。このグラフはモジュールを頂点集合とし、隣接するセルに存在するモジュール同士に辺を持つグラフである。このグラフが連結であるとき、モジュールは連結であると言う。本稿ではモジュールは常に連結であるとする。つまり、モジュールが移動する際に、移動後にモジュール同士の連結性が失われてしまうような移動は禁止する。（図 3）

3. 方角認識のあるモジュールロボットの探索アルゴリズム

まずは、モジュールに方角認識がある場合の探索アルゴリズムを示す。本章では 3 台のモジュールから成るモジュールロボットを想定し、各モジュールの視界は図 4 のように 12 近傍とする。

定理 1. 視界が 12 近傍である無記憶な、同期式のモジュール 3 台から成るモジュールロボットは任意の初期配置から探索問題を解くことができる。

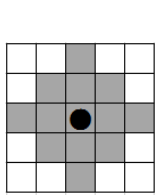


図 4 12 近傍の視界

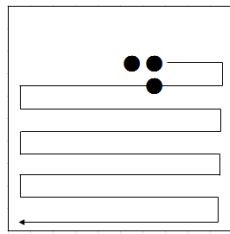


図 5 南端の角に至るまでの移動

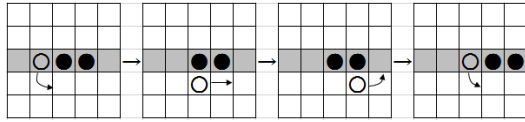


図 6 東方向への移動

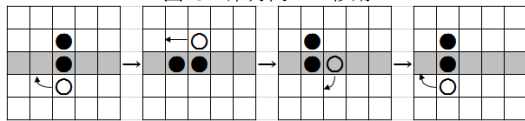


図 7 西方向への移動

以下では定理 1 を証明するために探索問題を解くアルゴリズムを示す。提案アルゴリズムではモジュールロボットが移動する際、ある行に沿って移動する。この行をベースラインと呼び、本稿の図では灰色のセルで示す。モジュールロボットは任意の初期配置から探索を開始し、図 5 のように東あるいは西の方向に移動し、壁に接すると南方向にベースラインを 1 段下げて方向転換を行うことで反対方向へ水平移動を行う。このベースラインを 1 段南に下げつつ東西へ往復する動作をスイープと呼ぶ。南端の行をスイープした後は移動方向を 90 度回転し、壁に沿って北側への移動を行い、北西あるいは北東の角に移動する。北東あるいは北西の角に到達すると再び最北の行からスイープ動作を行い、南西あるいは南東の角に到達するという動作を繰り返す。

探索の各場面における移動方法を示す。

- (1) 東方向への移動：図 6 のように、横一直線の配置から再び同じ配置に戻るまでの 3 回の移動を繰り返すことで東方向への移動を行う。
- (2) 西方向への移動：図 7 のように、縦一直線の配置から再び同じ配置に戻るまでの 3 回の移動を繰り返すことで西方向への移動を行う。
- (3) 東端の壁際における方向転換：図 8 のように方向転換を行い、西方向への移動に移行する。方向転換の際にベースラインが 1 段下がる。
- (4) 西端の壁際における方向転換：図 9 のように方向転換を行い、東方向への移動に移行する。東の壁際と同様に、方向転換の際にベースラインが 1 段下がる。

ベースラインが探索領域の南端より 1 つのセル分上にある時に壁際に到達した際はこの 2 通りの移動とは異なる移動を行う。西端の壁に接した場合は、壁沿いに北方向へ移動し、再び図 5 の移動に戻る。

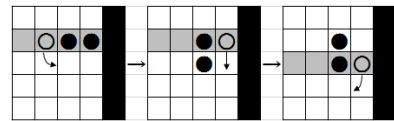


図 8 東の壁際における方向転換

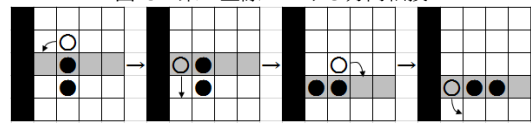


図 9 西の壁際における方向転換

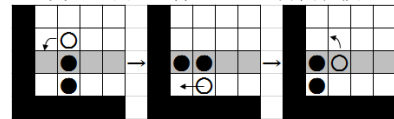


図 10 南西の角における方向転換

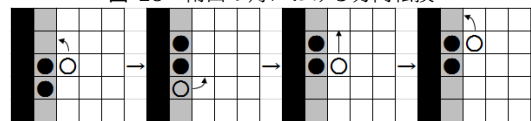


図 11 西端の壁際における北方向への移動

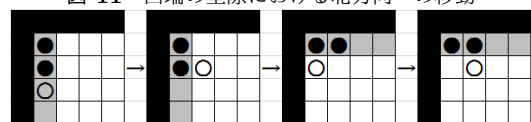


図 12 北西の角における方向転換

- (6) 南西の角における方向転換：図 10 のように方向転換を行い、北方向の移動への方向転換を開始する。
- (7) 西端の壁際における北方向への移動：北西の角へ到達するまで北方向へ図 11 のように移動を行う。北西の角に到達すると、図 12 のように東方向へ方向転換を行う。

以上の移動方法はモジュールロボットを構成する各モジュールがいずれの場合であるかを判定でき、しかもベースラインの移動をもとに、モジュールロボットが探索領域上の全てのセルを通ることが示せる。このように探索を行いながら、目標物を発見したモジュールがそのセルで停止することで、モジュールロボットは発見した目標物の位置を示す。

4. 方角認識のないモジュールロボットの探索アルゴリズム

次に、モジュールに方角認識がない場合の探索アルゴリズムを示す。本章では 5 台のモジュールから成るモジュールロボットを想定し、各モジュールの視界は図 13 のように 40 近傍とする。

定理 2. 視界が 40 近傍である無記憶な、同期式のモジュール 5 台から成るモジュールロボットは任意の初期配置から探索問題を解くことができる。

以下では定理 2 を証明するために探索問題を解くアルゴリズムを示す。方角認識のないモジュールロボットは、回転させて同じモジュール配置になってしまうようなモ

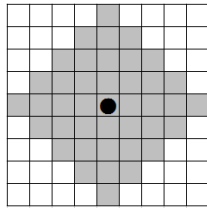


図 13 モジュール数 5 のロボットシステムのモジュールの視界範囲

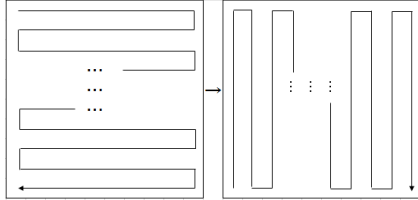


図 14 方角認識のないモジュール数 5 のモジュールロボットの移動の軌道

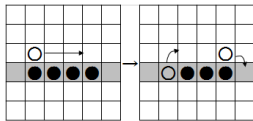


図 15 東方向への移動

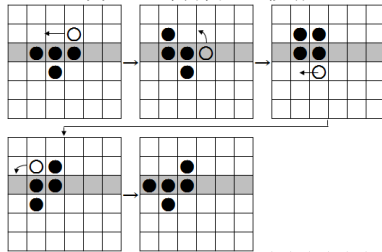


図 16 西方向への移動

ジュールロボットの形状を区別できないため、方角認識のあるモジュールロボットよりも取り得る配置の種類が少なくなってしまう。さらに、方角認識のないモジュールロボットを点対称な形状に配置すると、その対称性から他の点対称な形状に移動することしかできない、あるいはどのモジュールも移動できないか、いずれかの状態に陥ってしまう。そのため、探索アルゴリズム中では点対称な配置を回避しなくてはならない。提案アルゴリズムでは図 14 のように、東西あるいは南北のスイープ動作を繰り返し、探索範囲の角に到達すると 90 度方向転換し、同じ動作を繰り返す。つまり 4 通りの軌道があることになるが、回転すると全て同じ軌道になるため、このうち 1 つの軌道についての移動のみを考えれば良い。

図 14 の左の軌道について説明する。

- (1) 東方向への移動：壁が視界に入っていない状態で図 15 の移動を繰り返すことで実現する。
- (2) 西方向への移動：図 16 の移動を繰り返すことによって実現する。
- (3) 東の壁際における方向転換：図 17 のようにベースラインを変えずに方向転換を行う。
- (4) 西の壁際における移動方向転換：図 18 のように 1 段ベースラインが下がり東方向へ方向転換を行う。

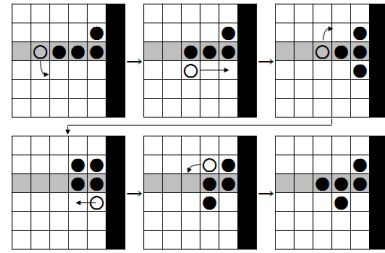


図 17 東の壁際における方向転換

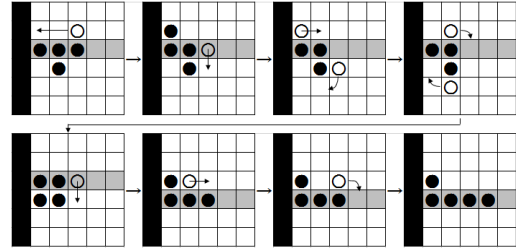


図 18 西の壁際における方向転換

西の壁際でのみベースラインが下がるため、南端の角における方向転換は、東西片側の角の方向転換のみ考えれば良い。本章で提案するアルゴリズムでは、南西の角において北方向へ方向転換を行う。方向転換後の最後の状態は図 15 の左の状態を反時計回りに 90 度回転させた状態である。南西での方向転換後は、図 14 の右の軌道で移動を行う。この軌道の南北の移動、壁際の方転換に関しても、上記の移動を反時計回りに回転させたものとなる。南東の角に到達すると、再び反時計回りに 90 度回転させた移動を行う。以降も角に到達しては反時計回りに 90 度回転させた移動を行うという動作を繰り返すことにより、探索領域内の全てのセルを通過することができる。

以上の移動方法はモジュールロボットを構成する各モジュールがいずれの場合であるかを判定できる。

5. おわりに

本研究では、二次元格子領域中に置かれた目標物をモジュールロボットが発見する探索問題を導入し、モジュールの方向認識と台数に着目した探索アルゴリズムを示した。今後の課題は、今回示した探索アルゴリズムの台数に関する最適性を示すこと、そして、3 次元の探索領域においてモジュールロボットが探索問題を解決するアルゴリズムを求めることである。

参考文献

- [1] A. Dumitrescu, I. Suzuki, and M. Yamashita, "Formations for fast locomotion of metamorphic robotic system", International Journal of Robotics Research, **23**, 6, pp.583–593, 2004.
- [2] F. Chen, Y. Yamauchi, S. Kijima, and M. Yamashita, "Locomotion of metamorphic robotic system based on local information", In Proceedings of SRDS 2014, pp.40–45, 2014.