

# 動的環境でのマルチエージェント巡回問題における効果的な領域形状のための分割手法

服部 克哉<sup>1,a)</sup> 菅原 俊治<sup>1</sup>

**概要:** 本研究では、複数のエージェントが空間を自ら分割して協調するマルチエージェント巡回問題において、環境が動的に変化する場合に、効果的な領域形状を実現し、作業を効率化する手法を提案する。大規模な空間における巡回は、単体のエージェントのみでは物理的、能力的限界があるため、複数のエージェントで協調する必要がある。また、一部エージェントの故障にも対応できるように、各エージェントの自律的行動が必要である。この問題に対して、領域を分割しエージェント毎に各分割領域を担当する手法がある。各分割領域のエージェントの作業量は公平であることが望ましいため、分割領域を動的に変化させ対応する。その際に、あるエージェントの領域が他エージェントの領域に分断される場合がある。領域の分断が多数発生すると、望ましい分割とは言えず巡回のコストが高くなりその効率を低下させることがある。他方、環境によっては、狭い通路の存在など作業量均一化に領域の分断が必要なこともある。そこで不要に分断された領域を他のエージェントに譲渡し、エージェントの不要な他領域の通過を抑える手法を提案した。評価実験により、提案手法が分断された領域の数を抑えた領域分割が得られることを示す。

**キーワード:** マルチエージェント, 巡回問題, 領域分割, 分業

## Improve efficiency of multi-agent patrolling by using efficient graph partitionin

### 1. はじめに

近年、ロボット技術において、センサーやアクチュエータ等のハードウェア、及びそれらを制御する情報技術等のソフトウェアの発達により多くの分野でロボットが活躍するようになった。その中で特にロボットの活躍が望まれる分野に、人間の日常的な作業の代行、及び人間の活動が困難な場所での活動がある。具体例として、前者は清掃や警備、後者は宇宙空間や災害発生地域での活動などがある。しかし、作業空間が広大であると、バッテリー容量、移動速度等の物理的、性能的限界により、単独のロボットで作業を行うことは現実的ではない。現在では、スマートフォン等に代表されるモバイル通信技術の発達により、ロボット間でのリアルタイムな通信が可能であるため、それを利用した複数台のロボットの協調による問題の解決手法が模索されている。

複数台のロボットによる協調作業の手法として、全てのロボットを1台のロボットまたはシステムで集中的に管理する手法があるが、環境の拡大に伴い管理のための計算量が増大すること、管理システムで不具合が起こると全てのロボットが機能停止することなどの問題がある。加えて、ロボットの台数が変化したとき、システム面でのバージョンアップが行われたときなどに、効果的な協調の構造が変化し、それらをあらかじめ予測することは困難である。このため各ロボットによる自律的な分担の学習が望まれる。

ロボットを始めとしたエージェントを複数用いて取り扱う問題の例としてマルチエージェント巡回問題がある。この問題では環境全体を均一的に巡回しながらも、作業効率化のために各エージェントの負担をなるべく公平にする分業が必要である。均一な作業分担の手法として、領域を分割し、各エージェントが分割された領域を割り当てられ巡回する手法がある。分割された領域は要求される巡回難度の差や形状、スロープや障害物の有無、エージェントの性能などを考慮し、同一の広さというよりはエージェントの作業量を均一とするべきである。これによって、エージェ

<sup>1</sup> 早稲田大学基幹理工学研究科, 東京都  
Fundamental Science and Engineering, Waseda University, 3-4-1  
Okubo, Shinjuku-ku, Tokyo, 169-8555 Japan

<sup>a)</sup> k.hattori@isl.cs.waseda.ac.jp

ント間の公平性を実現し、効率的に領域を巡回できる。マルチエージェント巡回問題の例として、警備の巡回や巡回清掃があり、ここでは清掃を例に述べる。

たとえば [1] の手法では、エージェントが自領域内のゴミの残量を推定し、その値の逆数を隣接した領域の担当エージェントと比較して、大きい方が隣接領域の一部へ担当領域を拡大することで、公平な分業を実現している。しかし、[1] の手法はあるエージェントの担当領域が他のエージェントの担当領域に分断されることがある。そのような場合、エージェントは他のエージェントの担当領域を頻繁に通過しなければ全ての担当領域を巡回できない。エージェントが他領域を通過することは、特定の場所への過剰な訪問や、実質的担当領域の大きさを超える巡回範囲など、作業量の増加を誘発する。これは公平となるように調整された作業量のバランスを崩す可能性がある。また他のエージェントの担当領域を通過すると、その領域を本来担当していたエージェントが回収するはずのゴミを回収し、そのエージェントが誤って実際よりゴミの量を少なく見積り、行動を変える事態も発生させる。しかし、狭い通路や入口など環境の形状によっては領域の分断が必須なこともある。

これを考慮に入れ、不要な領域の分断と他領域の通過を抑えた領域分割配置の手法として [2] がある。[2] では、その手法を [1] の手法と比較する実験を行い、不要な領域の分断と他領域の通過を抑えると巡回の効率が改善することを示している。しかし、[2] の実験では環境が常に一定であったため、本研究では、動的な環境の下で実験を行うことで環境の変化に適応した巡回が行えるか検証する。

## 2. 関連研究

先行研究において、継続的な警備や清掃を行う際の課題を解決する手法には大きく分けて 2 つある。第 1 は作業を行う領域全体をエージェントが共有して清掃する手法である。たとえば、[3] ではエージェントが様々な確率で発生するイベントを、発生確率の高い場所から優先的に巡回し、回収する。しかし、複数エージェントの協調は考慮されていない。[4] では時間の経過とともに環境構造が変化の中で、あるマスターエージェントが各エージェントの移動先を判断している。しかし、これは集中的な制御手法であり、マスターエージェントが機能しなくなった際に、その影響がエージェント全体に及ぶ。[5] ではエージェントにバッテリーのモデルを導入し、バッテリー残量と他のエージェントの探索状況から探索を継続するか充電に戻るかを決定している。また、[6] では、エージェントが強化学習で探索アルゴリズムを決定する。[5]、[6] の手法では領域分割がされていないため、エージェント間の作業領域の重複が発生し、効率が低下する可能性がある。

第 2 の手法は環境全体の領域を分割し、各エージェントが自らの担当となる領域を清掃する手法である。[7] は [3]

の手法をマルチエージェントに拡張し、エージェント間で互いの領域の境界に関する情報を通信で交換し、重なりがあるとその領域を頻繁に訪れているエージェントに譲渡することで領域分割を実現している。[8] では各エージェントの担当領域をガスで満たされた風船と見立てる Ballon モデルとアリのフェロモンモデルを用いて、各エージェントの担当領域の膨張力（担当領域の広さ）が均衡するように領域を分割している。しかし、[7]、[8] はどちらもエージェントのバッテリーによる制約を考慮していない。また、[8] は各担当領域の広さが等しくなるように分割しているだけで汚れやすさの偏りや領域の形状などを考慮していない。[1] ではバッテリー容量に制約を持つエージェントが担当領域内のゴミの残量の推定値の逆数を拡張力として領域を分割している。しかし、この手法では各エージェントの担当領域が他のエージェントの担当領域によって分断され、効率低下を招くこともある。また、逆に領域が分断されることが必須である環境も考慮していない。[9] ではメンバーの探索領域をリーダーが決定するが、リーダーが故障したとき全エージェントが影響を受ける。[10] では障害物がある環境でバッテリーに制限がある複数のエージェントが、領域を分割して清掃を行う。しかし、この手法ではある分割領域が他の分割領域に分断される場合を考慮していない。本研究では [1] を拡張し、不要な領域の分断を減少させ、効率向上を目指している。

## 3. モデルの定義

この節では本研究で使用する環境とエージェントのモデル、領域分割手法を説明する。

### 3.1 環境のモデル

$A = \{1, \dots, N\}$  をエージェントの集合とする。時間は離散時間で、最小単位をステップとする。エージェントは 1 ステップで移動とゴミの回収を行う。エージェントが巡回清掃を行う環境を連結グラフ  $G = (V, E)$  で表す。ここで  $V = \{v_1, \dots, v_x\}$  はノードの集合、 $E$  はエッジの集合とする。ノード  $v_i, v_j \in V$  間に存在するエッジを  $e_{i,j}$  と表す。本研究では、全エッジの重み（長さ）は 1 であり、エージェントは各ノードからエッジの存在する隣接ノードへ 1 ステップで移動可能とする。エージェント  $i$  はそれぞれ定められた位置に充電基地  $v_{base}^i (\in V)$  があり、そこで充電をする。

ある時刻  $t$  にエージェント  $i$  は担当領域を持ち、サブグラフ  $G_t^i = (V_t^i, E_t^i)$  ( $V_t^i \subseteq V, E_t^i \subseteq E$ ) で表される。各エージェントの作業量を均一化するため、エージェント  $i$  の担当領域の大きさ  $|V_t^i|$  はゴミの清掃状況によって変化する。

### 3.2 ゴミ発生・蓄積モデル

清掃を行う環境内では、各ノード  $v$  において 1 ステップ

毎に確率  $p_v (0 \leq p_v \leq 1)$  でゴミが発生する。ゴミは蓄積し、ノード  $v$  の時刻  $t$  におけるゴミの量  $L_t(v)$  は、

$$L_{t+1}(v) = \begin{cases} L_t(v) + 1 & (\text{確率 } p_v) \\ L_t(v) & (\text{その他}) \end{cases} \quad (1)$$

と表される。エージェントが時刻  $t$  にノード  $v$  へ到達すると、そのノードのゴミは清掃されたとして  $L_{t+1}(v) = 0$  とする。

### 3.3 エージェントのモデル

エージェントのモデルは、次の3つを想定する。第1にエージェントが清掃を行う空間のマップは既知とする。今まで多くのマップ作成のアルゴリズム [11][12] が提案されている。本研究では領域分割に主眼を置くため、将来的にこれらのアルゴリズムを活用しマップを既知にできると仮定する。第2に複数台のエージェントが同一のノードに存在可能で、衝突は考えない。現実では複数台のエージェントが同一地点に移動しようとする時衝突が発生する。しかし、ここでは領域分割に主眼を置くため、既存の衝突回避アルゴリズムで衝突を回避できると仮定し、考慮から外した。第3に各エージェントは有限容量のバッテリーを保持する。そのためエージェントは無限に探索を続けることはできず、探索状態と充電状態という2つの状態を交互に繰り返す。なお、今回は簡単のため通信によるバッテリーの消費は考慮しない。

#### 3.3.1 エージェントの状態

エージェントの状態には探索状態と充電状態の2つがある。探索状態とは、エージェント  $i$  が清掃のためグラフを探索している状態である。このときエージェントは各々の探索アルゴリズムにしたがって、自身の担当領域を清掃する。 $i$  の1ステップ毎のバッテリー消費量を  $B_{drain}^i$ 、時刻  $t$  でのバッテリー残量を  $B_t^i$  とすると、時刻  $t+1$  でのバッテリー残量は

$$B_{t+1}^i = B_t^i - B_{drain}^i \quad (2)$$

となる。よってエージェント  $i$  の連続稼働時間の最大は、バッテリーの最大値  $B_{max}$  を用いて  $B_{max}^i / B_{drain}^i$  ステップと表せる。連続稼働時間が有限のため、エージェントは移動時に充電量が0を下回るノードへは到達できない。エージェント  $i$  がノード  $v$  から基地まで移動するときに必要なエネルギー  $Pot^i(v)$  は、 $Len(v, u)$  をノード  $v$  からノード  $u$  への最短距離として、

$$Pot^i(v) = Len(v, v_{base}^i) \times B_{drain}^i \quad (3)$$

と表される。そのため現在のノード  $v$  から担当領域内のあるノード  $v^i$  に移動する場合、

$$B_t^i < Pot^i(v) + Len(v^i, v) \times B_{drain}^i \quad (4)$$

という条件を満たすノードには到達できない。このときエージェントはノード  $v$  へ移動せず、最短経路で充電基地へと帰還する。

次に、充電状態とはエージェント  $i$  が自身の充電基地  $v_{base}^i$  で充電をしている状態のことである。充電にかかる時間  $t_{charge}^i$  はバッテリー消費量に比例すると仮定し、比例定数  $k_{charge}^i$  を用いて

$$t_{charge}^i = k_{charge}^i (B_{max}^i - B_t^i) \quad (5)$$

と表される。簡単のため、本研究においてエージェントは満充電してから探索を再開する。

### 3.4 初期担当領域

エージェント  $i$  は充電基地  $v_{base}^i$  から距離  $d_{init}$  以下の範囲の空間を初期担当領域  $G_0^i = (V_0^i, E_0^i)$  として持つ。本論文における距離とは、ノード間の最短経路を辿ったときに通過する全てのエッジの重みの和である。また充電基地  $v_{base}^i$  はいかなる場合でも  $v_{base}^i \in V_t^i$  と仮定する。

### 3.5 飛び地

本研究における飛び地とは次のように定義される。エージェント  $i$  の時刻  $t$  における担当領域  $G_t^i = (V_t^i, E_t^i)$  内のあるノード  $v \in G_t^i$  に関して、 $v$  から  $G_t^i$  に含まれるノードとエッジのみを通過して辿り着ける領域を連結領域  $G_{enc_j}^i$  とし、これを飛び地と呼ぶ。 $i$  の連結領域  $G_{enc_j}^i$  の集合  $\mathcal{G}_{enc}^i$  を飛び地の集合とする。連結領域は共通のノードをもたない。飛び地の集合  $\mathcal{G}_{enc}^i$  の大きさ  $|\mathcal{G}_{enc}^i|$  を担当領域の飛び地の数と呼ぶ。なお、 $|\mathcal{G}_{enc}^i| \geq 1$  である。

### 3.6 通過領域

本研究における通過領域を次のように定義する。時刻  $t$  のエージェント  $i$  の担当領域  $G_t^i$  について、 $G_t^i$  に含まれる全ノードを巡回する際に通過が必要な、他エージェントの担当領域のノードの集合である。通過領域を決定するアルゴリズムの詳細は第4.2節で述べる。

### 3.7 エージェントの行動モデル

#### 3.7.1 探索アルゴリズム

エージェントはある探索アルゴリズムを用いて担当領域内のノードを巡回する。本研究では領域分割に主眼を置くため、各エージェントの探索アルゴリズムは有向深さ優先探索 (DDFE) に固定する。以下でDDFEのアルゴリズムを簡単に説明する。

エージェント  $i$  は充電基地を出発するとき、担当領域内のノードの中でゴミの量の推定値  $E(L_t(v))$  が最も大きなノードをターゲットノードと設定し、最短経路で移動する。ターゲットノードに到達したら、充電基地出発後に未訪問の隣接ノードをランダムに1つ選択して移動し、そのノード

ドをスタックの上部にプッシュする。エージェント  $i$  が未訪問のノードを選び続けられる限りこれを繰り返す。エージェント  $i$  は選択できるノードが存在しないとき、スタックの一番上のノードをポップしてそのノードに移動する。その後、エージェント  $i$  は再び未訪問のノードを選択する。この動作を繰り返して最初に設定したターゲットノードへと戻ってきた場合、基地に最短経路で戻る。なお、ターゲットノードと充電基地間の移動では、領域の担当を考慮せず最短経路で移動するため、他のエージェントの領域を通過することもある。

### 3.7.2 領域拡大行動の開始

領域拡大行動とは、どのエージェントの担当領域にも含まれない領域、及び作業量が多い他のエージェントの担当領域に対して、自らの担当領域の拡大を試みることである。エージェントが領域拡大行動を試みる条件を次に示す。エージェント  $i$  は時刻  $t_b$  に充電基地  $v_{base}^i$  を出発する際に、自らの担当領域内における、 $t_b$  から  $\gamma$  ステップ後のゴミの残量の推定値  $E(L_{t_b+\gamma}(V_{t_b}^i))$  を計算する。時刻  $t_b$  ではなく時刻  $t_b + \gamma$  でのゴミの残量の推定値を計算するのは、基地出発からある程度時間が経過したとき領域拡大の試みが行われるためである。

基地出発時のゴミの残量の推定値をもとに領域拡大を試みようとする、実際状況とは異なる推定値で領域拡大を行う可能性がある。これを防ぐために未来の残量で推定値を計算している。エージェント  $i$  は時刻  $t_b$  に巡回を開始したとき、時刻  $t (> t_b)$  までに自分が訪れたノードの数  $N_{vis}(t)$  と回収したゴミの総和  $N_d(t)$  を記録する。そして下記の2つの条件式

$$N_{vis}(t) \geq R_1 \cdot |V_t^i| \quad (6)$$

$$N_d(t) \geq R_2 \cdot E(L_{t_b+\gamma}(G_{t_b}^i)) \quad (7)$$

を満たしたとき担当領域の拡大を試みる。このとき  $R_1, R_2$  はあらかじめ定めたパラメータであり  $0 \leq R_1, R_2 \leq 1$  である。過剰な領域拡大行動を避けるため、領域拡大を試みるのは充電基地を出発してから帰還するまでの間に1回のみとしている。

また上記の領域拡大行動以外に、エージェントは基地出発時、担当領域に隣接する領域でどのエージェントの担当領域でもない領域に担当領域を拡大する。これは、各エージェントの担当領域の和が空間全体を覆うまでの時間を短縮するためのものである。

### 3.7.3 拡大戦略

ここではエージェントの領域拡大戦略について述べる。エージェント  $i$  は領域拡大を試みる際に、対象となるノードを以下のように決定する。

- (1) エージェント  $i$  は自身の担当領域の隣接ノードの内、自身の担当領域に含まれないノードの集合  $B$  を選択

する。

- (2) エージェント  $i$  は  $B$  の中から、 $I_{avoid}^i$  (これは3.7.5で説明する) に含まれず、自分の基地  $v_{base}^i$  から最も距離の小さい  $k_{inc} (> 0)$  個のノードを選び、その集合を  $I_{inc}^i$  とする。

- (3) ノード  $I_{inc}^i$  とそれらの隣接ノードの集合のうち、 $V_t^i$  にも  $I_{avoid}^i$  にも含まれないものを  $I^i$  とする。

$I^i = \emptyset$  であるとき  $i$  は領域拡大行動を行わない。

### 3.7.4 拡大力

領域拡大対象となったノードを担当領域に加えるかどうか決定する際に、エージェントがどの程度既存の担当領域内を清掃できているかを示す拡大力 (expansion power) を比較する。拡大力は次のように定義する。エージェント  $i$  は充電基地へ帰還すると、その時刻  $t$  における担当領域内のゴミの残量の推定値  $E(L(V_t^i))$  を求める。そのとき  $i$  の拡大力  $\varepsilon(i, t)$  は  $E(L(V_t^i))$  を用いて

$$\varepsilon(i, t) = E(L(V_t^i))^{-1} \quad (8)$$

とする。ただし  $E(L(V_t^i)) = 0$  の場合  $\varepsilon(i, t)$  は十分に大きな値とする。各エージェントは充電基地に帰還し拡大力を再計算するまでこの値を保持する。

### 3.7.5 領域拡大のための交渉

エージェント  $i$  は拡大対象のノード  $I^i$  に対して、そのノードを現在担当領域に持っているエージェント  $j$  の拡大力  $\varepsilon(j, t)$  と自らの拡大力  $\varepsilon(i, t)$  を比較する。 $\varepsilon(i, t)$  の方が大きいときには、そのノードを  $i$  の担当領域に加え、もともと担当していたエージェント  $j$  の担当領域から削除する。 $\varepsilon(i, t)$  の方が小さいときには、そのノードへの拡大は失敗したとして担当領域は変化しない。交渉後は  $I^i$  に含まれているノードの内、拡大に失敗したノードの集合を  $I_{avoid}^i$  に加え、 $k_{avoid}$  回の間は領域拡大の対象となる  $I^i$  の要素に選ばれないとする。これは同じノードに対する領域拡大行動の頻繁な失敗を避けるためである。

## 3.8 システムの評価基準

本研究のシステムの評価方法を述べる。本研究では、飛び地の数や通過領域の大きさを減少させた効果的な領域分割を行うことを目的とするため、各エージェントの飛び地の数と通過領域の大きさ、全エージェントの通過領域の大きさを評価する。また領域分割の結果効率的な巡回が行われているか確かめるために、以下の式で表される、ある期間  $[t_s, t_e]$  での全ノードのゴミの残量の総和の平均値  $D_{t_s, t_e}(V)$  を評価する。

$$D_{t_s, t_e}(V) = \sum_{t=t_s}^{t_e} L_t(V) / (t_e - t_s + 1) \quad (9)$$

さらに、各エージェントの作業量が均一であることを確認するために、それぞれの担当領域の大きさと担当領域内のゴ



ミの残量の総和の平均  $L_t(V_t^i)$  を評価する.

## 4. 提案手法と評価実験

今回行う実験は [1] と [2] を拡張したものであり, 特定のグラフ構造において担当領域のグラフが複数の飛び地に分かれる現象を緩和するとともに, シミュレーションの途中で環境が変化した場合の検証を行う.

### 4.1 飛び地発生緩和

エージェント  $i$  は飛び地の発生を緩和するため, 充電を終えて基地を出発する際に, 次に示す条件を満たした連結領域  $G_{enc_j}^i$  を, 他のエージェントと通信による交渉を行い譲渡する.  $i$  は時刻  $t$  に充電基地を出発するとき, 担当領域  $G_t^i$  内に存在する各連結領域  $G_{enc_j}^i \in \mathcal{G}_{enc}^i$  に対して次の評価値  $E_{enc_j}^i$  を計算する. ここで  $dis^i(v)$  はエージェント  $i$  の充電基地  $v_{base}^i$  からノード  $v$  までの距離である. なお, 距離は第 3.4 節で定義したものをを用いる.

$$E_{enc_j}^i = \frac{1}{|G_{enc_j}^i|} \sum_{v \in G_{enc_j}^i} dis^i(v) \quad (10)$$

$E_{enc_j}^i$  と担当領域の大きさ  $|G_t^i|$  が定数  $R_3$  を用いて条件式

$$E_{enc_j}^i > \frac{|G_{enc_j}^i|}{R_3 \cdot |G_t^i|} \quad (11)$$

を満たすとき, その連結領域  $G_{enc_j}^i$  を譲渡する対象に選択する.  $R_3$  はあらかじめ定めたパラメータとする. なお, この条件を満たす  $G_{enc_j}^i$  が複数存在する場合は, その全てを譲渡の対象とする. ただし,  $G_{enc_j}^i$  が充電基地を含む場合は譲渡の対象としない. 譲渡の対象となるエージェントは, 担当領域に, 譲渡する連結領域  $G_{enc_j}^i$  に隣接するノードが最も多いエージェントである.

他方, 飛び地を譲渡されたエージェントは, それを自らの担当領域に加える. これにより, 飛び地を譲渡されたエージェントの担当領域が他のエージェントの担当領域より大きくなるが, 担当領域の拡大により式 8 の拡大力が減少し, 他のエージェントによる領域拡大を受けやすくなるため, 結果的にエージェント間における作業量の公平性は保たれる.

### 4.2 通過領域の決定と担当領域の巡回

エージェント  $i$  は自身の担当領域が拡大または縮小したときに, 変化後の担当領域から巡回のための通過領域を定める. ここでは飛び地間の距離を, 各飛び地に含まれる任意のノードの組み合わせの中で, 2 ノード間の距離が最小となる組み合わせ  $(v_s, v_t)$  の距離と定義する. また飛び地間の最短経路とは  $(v_s, v_t)$  の最短経路と定義する. 通過領域  $G_{ins}^i$  を次の通り定める. ただし  $G_{ins}^i$  は初期値として  $G_{ins}^i = \phi$  とおく.

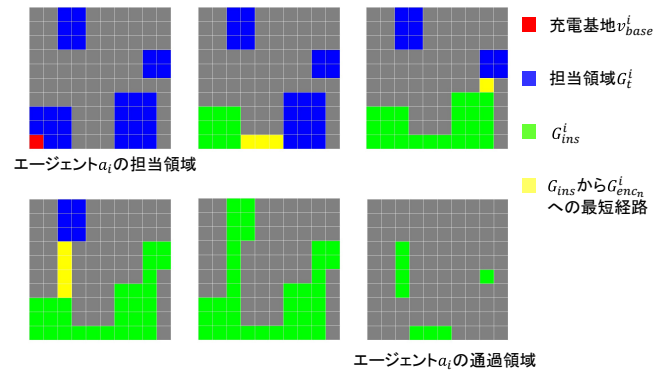


図 1: 通過領域決定アルゴリズム

- (1)  $i$  はノードの集合  $G_{ins}^i$  に自身の充電基地  $v_{base}^i$  を含む連結領域  $G_{enc_{base}}^i$  のノードを加える.
- (2)  $G_{ins}^i$  に含まれていない連結領域  $G_{enc_j}^i$  の中で, 距離が最小のものを 1 つ選び, それを  $G_{enc_n}^i$  とする.
- (3)  $G_{ins}^i$  に,  $G_{enc_n}^i$  に含まれるノード, 及び  $G_{ins}^i$  と  $G_{enc_n}^i$  間の最短経路上のノードを加える.
- (4)  $G_{ins}^i$  に含まれないノードを持つ連結領域  $G_{enc_j}^i$  が存在しなくなるまで (1)~(3) を繰り返す.
- (5)  $G_{ins}^i$  から  $i$  の担当領域に含まれるノードを全て取り除いたものを  $i$  の通過領域とする.

エージェント  $i$  は担当領域を巡回する際に, 通過領域も担当領域の一部とする.

## 5. 評価実験

### 5.1 実験環境

提案手法の性能を動的環境で調査するために次の実験を行った. エージェントが清掃を行う空間  $G$  は図 2 に示す 2 次元のグリッドグラフとする. エージェント  $i \in A = \{1, 2, 3, 4, 5\}$  は充電基地  $v_{base}^i$  を出発し  $G$  全体を巡回清掃する. ゴミの発生確率を一様に  $p_h = 2 \times 10^{-5}$  とする. シミュレーションは 200 万ステップで行われ, 100 万ステップが経過するとエージェントの台数が変化する. 実験 1 では, 100 万ステップ経過後に新しいエージェントが 1 台追加され, 実験 2 では, それまで巡回していた内の 1 台が停止する.

実験に用いるエージェントの各種パラメータの設定を表 1 に示す. 実験では, 3600 ステップ毎の

- (1) 全ノードにおけるゴミ残量の総和の推移
- (2) 各担当領域のゴミ残量の総和の推移
- (3) 各担当領域の飛び地の数の推移
- (4) 全エージェントの通過領域の大きさの和の推移

について調査する. 実験結果のデータは上記の実験を 30 回行った平均値である. 比較対象となる既存手法として, 先行研究 [1] の手法に, 通過領域の決定と巡回清掃中の通

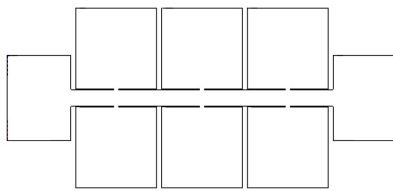


図2: エージェントが清掃を行う空間  $G$

表 1: 実験設定

	パラメータ	値
エージェント台数	$ A $	35 台
拡大戦略を行うためのパラメータ	$R_1$	0.4
	$R_2$	0.4
	$R_3$	0.005
	$\gamma$	300
拡大戦略の制御パラメータ	$k_{avoid}$	90
バッテリー消費量	$B_{drain}$	1
バッテリー最大容量	$B_{max}$	900
充電にかかる時間の比例定数	$k_{charge}$	3
初期担当領域の範囲	$d_{ini}$	2

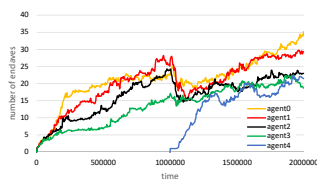


図3: 実験 1 各担当領域の飛び地の数 [既存手法]

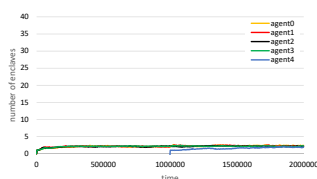


図4: 実験 1 各担当領域の飛び地の数 [提案手法]

過領域の通行を加えたものを採用した。

## 5.2 実験結果と考察

### 5.2.1 実験 1

100 万ステップ経過時にエージェントを 1 台追加した場合の評価実験の結果として、まず各担当領域の飛び地の数の推移を図 3, 4 に示す。100 万ステップまでは、既存手法は時間の経過とともに飛び地の数が増加するのに対し、提案手法は飛び地の発生を抑制した状態を維持している。これは図 6 に示すゴミ残量の総和の推移にも表れているよう

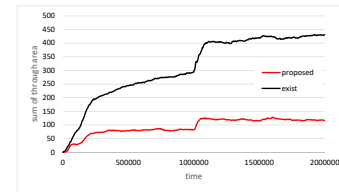


図5: 実験 1 全エージェントの他領域通過量の和の推移

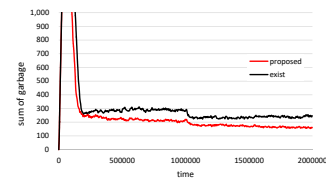


図6: 実験 1 全ノードのゴミ残量の総和

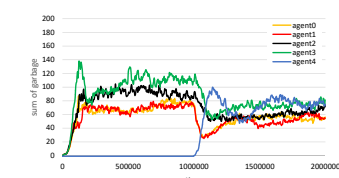


図7: 実験 1 各担当領域のゴミ残量 [既存手法]

に、既存手法と比べて効率の向上が確認できる。また図 5 に示す他領域の通過量も、既存手法では、時間とともに増加し、これが効率の激減となる (図 6) が、提案手法では、これが抑えられ、むしろ学習の進行にともない、効率はやや向上する (図 6)。

100 万ステップ経過後も、既存手法では (図 3)、エージェントが追加された直後は、追加されたエージェントが周りの飛び地を取り込みながら担当領域を拡大するため、一時的に飛び地の数が減少する。しかし、その後時間の経過とともに徐々に飛び地の数が増加している。それに対して提案手法 (図 4) では、エージェントが追加されても飛び地の数にほとんど変化がない。

各手法における全エージェントの通過領域の和を比較したものを図 5 に示す。100 万ステップ経過までは、提案手法の方が既存手法と比べ通過領域の拡大を抑えられている。これは飛び地の発生の抑制により、他領域を通過する頻度が減少したためと考えられる。100 万ステップ経過後は、どちらも新規エージェント追加により領域分割が細分化したため通過領域の和が増加したが、提案手法の方が通過領域小さく保たれる。実験 1 の領域分割の例を示した図 15 からもこれは分かる。これらより、提案手法がエージェントの追加により環境が変化したとしても、飛び地の発生と通過領域の拡大を抑制した領域分割が実現できたと思われる。

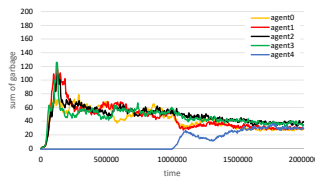


図 8: 実験 1 各担当領域のゴミ残量 [提案手法]

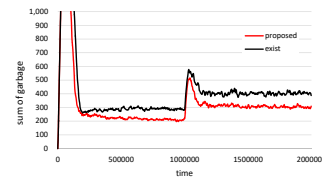


図 12: 実験 2 全ノードのゴミ残量の総和

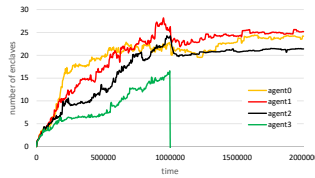


図 9: 実験 2 各担当領域の飛び地の数 [既存手法]

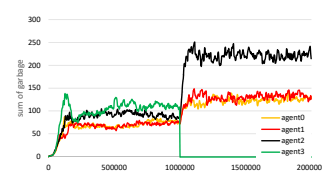


図 13: 実験 2 各担当領域のゴミ残量 [既存手法]

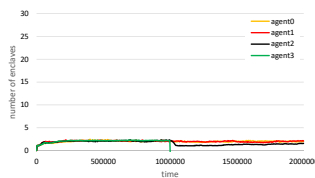


図 10: 実験 2 各担当領域の飛び地の数 [提案手法]

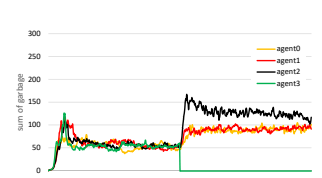


図 14: 実験 2 各担当領域のゴミ残量 [提案手法]

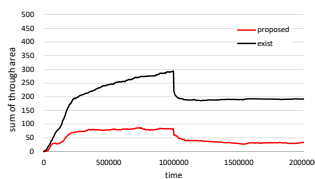


図 11: 実験 2 全エージェントの他領域通過量の和の推移

図 6 で環境全体のゴミの残量の推移を、図 7, 図 8 で各手法におけるそれぞれの担当領域内のゴミの残量の推移を示す。100 万ステップ経過までは、提案手法が既存手法と比べ、環境全体のゴミの残量を少なく保て、各エージェントの担当領域内のゴミの残量もより均一化できた。100 万ステップ経過後は、両手法ともエージェント台数の増加により、環境全体のゴミの残量と各担当領域のゴミの残量が減少した。また、提案手法はエージェントが追加された後も既存手法より環境全体のゴミの残量を少なく保て、各担当領域のゴミの残量の均一化ができた。これらから、提案手法が通過領域や飛び地の増加を抑え、効率的な巡回ができたことが分かる。

### 5.2.2 実験 2

実験 1 とは逆に 100 万ステップ経過後に稼働中のエージェント 1 台を停止させた場合の実験結果を示す。なお、100 万ステップ経過までの結果は実験 1 と同様である。各担当領域の飛び地の数を図 9, 10 に示す。エージェント 1

台が停止した後でも、提案手法は既存手法と比べて飛び地の発生を抑制できている。図 11 では各手法における全エージェントの通過領域の和を示している。どちらの手法もエージェント 1 台が停止後は、エージェント数の減少による分割領域の拡大で通過領域の大きさが減少している。また、エージェント 1 台が停止した後でも提案手法は既存手法よりも全体的に他領域の通過量は小さい。実験 2 の領域分割の例を示した図 16 からこれが分かる。これより提案手法は、実験 1 と同様に環境が変化したとしても、飛び地の発生と通過領域の拡大を抑制した領域分割ができたことが分かる。

図 12 で環境全体のゴミの残量の推移を示す。どちらの手法でも、エージェント 1 台が停止した直後に、一時的に担当のいない領域が発生するためゴミの残量が増加している。しかし、領域拡大により停止したエージェントの領域を他のエージェントが補うことで、エージェント停止前ほどではないがゴミの残量は減少している。また、実験 1 と同様に提案手法が既存手法と比べて環境全体のゴミの残量を少なく保てたことが分かる。このことから、通過領域の拡大や飛び地の増加を抑えて、効率的な巡回が行えたことが分かる。

図 13, 図 14 で各手法におけるそれぞれの担当領域内のゴミの残量の推移を示す。既存手法、提案手法ともにエージェントが停止した後は各担当領域のゴミの残量が増加し

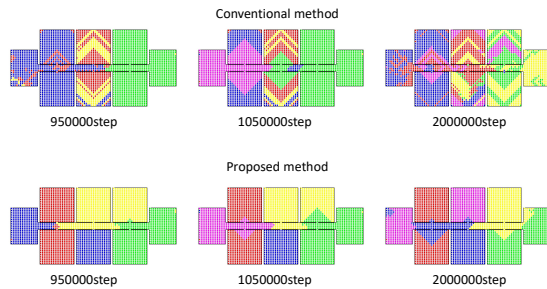


図 15: 実験 1 領域分割の例

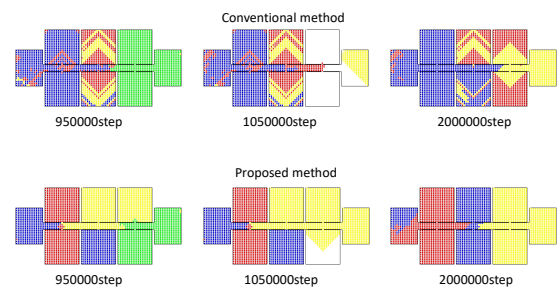


図 16: 実験 2 領域分割の例

ており、特に停止したエージェントと充電基地が近いエージェントの担当領域でその傾向が顕著となっている。しかし、提案手法では徐々に各担当領域のゴミの残量が均一化されていることが分かる。これより、提案手法がエージェントの停止により環境が変化した後も、公平な分業を実現するための適応を行えていることが分かる。

## 6. まとめ

本研究では、マルチエージェントの巡回清掃問題における領域分割を用いた手法について、ある領域が他の領域を不要に分断することを抑制し、より効果的な領域分割配置を行う手法を示した。その手法が動的な環境においても適用可能か検証するために、エージェント数が増加する場合と減少する場合の2通りの環境で評価実験を行った。評価実験の結果、提案手法はどちらの場合においても、環境の変化に合わせた領域分割で分割領域の不要な分断を抑制し、効果的な領域分割配置の実現への有効性を示した。

今後の課題として、ゴミ発生確率を未知とすること、異なる形状の空間における今回の手法の有効性を検証することが挙げられる。今回の実験ではゴミの発生確率は既知としたが、実際の環境では未知な場合がほとんどである。この手法ではゴミの発生確率の推測値が領域拡大の際のパラメータの1つであるため、それを未知とし、学習により推測した際の影響を調査する必要がある。また今回の空間の形状は単純な長方形の組み合わせだが、現実の空間の形状はより多くの種類の形状の組み合わせとなっている。そのような空間でも提案手法が有効かを検証する。

謝辞：本研究はJSPS 科研費 17KT0044 の助成を受けたものです。

## 参考文献

[1] Kato, C. and Sugawara, T.: Decentralized Area Partitioning for a Cooperative Cleaning Task, *Proceedings of the 16th International Conference on Principles and Practice of Multi-Agent Systems (PRIMA-2013)*, pp. 470–477 (2013).  
[2] 服部克哉, 菅原俊治: マルチエージェント巡回問題における効果的な領域分割配置による効率化, 合同エージェント

ワークショップ&シンポジウム 2019 (JAWS2019) (2019).  
[3] Ahmadi, M. and Stone, P.: Continuous area sweeping: A task definition and initial approach, *Proceedings of 12th International Conference on Advanced Robotics (ICAR 2005)*, IEEE, pp. 316–323 (2005).  
[4] Othmani-Guibourg, M., Fallah-Seghrouchni, A. E., Farges, J. L. and Potop-Butucaru, M.: Multi-agent patrolling in dynamic environments, *2017 IEEE International Conference on Agents (ICA)*, pp. 72–77 (online), DOI: 10.1109/AGENTS.2017.8015305 (2017).  
[5] 米田圭佑, 加藤千紘, 菅原俊治: バッテリー制限付きマルチロボットによる継続的な巡回清掃における行動計画法の提案とその評価, エージェント合同シンポジウム (JAWS2012) 論文集, pp. 24–26 (2012).  
[6] Yoneda, K., Kato, C. and Sugawara, T.: Autonomous Learning of Target Decision Strategies without Communications for Continuous Coordinated Cleaning Tasks, *IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology (WI-IAT 2013)*, pp. 216–223 (2013).  
[7] Ahmadi, M. and Stone, P.: A multi-robot system for continuous area sweeping tasks, *Proceedings of 2006 IEEE International Conference on Robotics and Automation (ICRA 2006)*, pp. 1724–1729 (2006).  
[8] Elor, Y. and Bruckstein, A.: Multi-agent graph patrolling and partitioning, *Proceedings of the 2009 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology-Volume 02, IEEE Computer Society*, pp. 52–57 (2009).  
[9] Nasir, A., Salam, Y. and Saleem, Y.: Multi-level decision making in hierarchical multi-agent robotic search teams, *The Journal of Engineering*, Vol. 2016, No. 11, pp. 378–385 (online), DOI: 10.1049/joe.2016.0076 (2016).  
[10] Sea, V., Kato, C. and Sugawara, T.: Coordinated Area Partitioning Method by Autonomous Agents for Continuous Cooperative Tasks, *Journal of Information Processing*, Vol. 25, pp. 75–87 (online), DOI: 10.2197/ipsjip.25.75 (2017).  
[11] Hahnel, D., Burgard, W., Fox, D. and Thrun, S.: An Efficient FastSLAM Algorithm for Generating Maps of Large-Scale Cyclic Environments from Raw Laser Range Measurements, *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003)*, vol. 1, pp. 206–211 (2003).  
[12] Wolf, D. F. and Sukhatme, G.: Mobile Robot Simultaneous Localization and Mapping in Dynamic Environments, *Autonomous Robots 19(1)*, pp. 53–65 (2005).