

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

服部 克哉¹ 杉山 歩未¹ 菅原 俊治¹

概要: 本研究では、複数のエージェントが領域を自律的に分割して協力し合うマルチエージェント巡回問題において、不要な領域の分断を抑え、作業を効率化する手法を提案する。大規模な空間の巡回問題では、ロボットなどのエージェントの物理的、能力的限界から、複数のエージェントの協調巡回が必要である。また、一部のエージェントが故障しても対応できるように、各エージェントの自律的行動が求められる。複数のエージェントが協調して巡回を行う際の手法に、領域をボトムアップに分割し、各分割領域をそれぞれのエージェントが担当する手法がある。この手法では各エージェントの担当領域内での作業量が均一となる分割が必要だが、領域を環境の状態に応じて動的に変化させる際に、あるエージェントの担当領域が他の担当領域によって分断されることもある。この分断された領域が多数発生すると、エージェントは自担当領域を巡回するために、他のエージェントの担当領域を通過する必要がある、これが効率を低下させることがある。他方、領域の形状や充電基地の設置条件によっては、負荷分散の観点から領域の分断が必須なこともある。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する交渉を加え、エージェントの無駄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、他領域の通過を抑制する効果的な領域分割が得られることを示す。

1. はじめに

近年ではセンサーやアクチュエータ、それらを制御する情報技術、ロボット技術の発達にともない、様々な分野でロボットが活躍するようになった。ロボットの活躍が特に望まれる分野として、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。具体的には、前者の例として清掃や警備、後者は宇宙空間や災害発生地域での活動が挙げられる。しかし1台のロボットのみでの作業には、バッテリー容量や移動速度などの物理的制約や性能面の限界があり、特に環境が広いときには現実的ではない。現在は無線通信技術の発達から複数台のロボット間での通信が可能となり、それを利用した複数台のロボットによる協調作業での問題の解決方法が模索されている。

複数台のロボットの協調作業を実現するために、集中的に全てのロボットを管理する手法があるが、環境が大規模化すると管理のための計算量が膨大になること、管理システムが機能しなくな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。このため各ロボットが自律的に行動することが望まれる。

複数のロボットによる協調を用いて取り扱う問題の例として巡回清掃問題がある。この問題において環境全体を均一的に清潔に保ちながらも、各ロボッ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、分割された領域を各エージェントに割り当てそこを巡回する手法がある。分割された領域は汚れやすさや領域の形状、スロープの有無、エージェントの性能などを考慮し、同一の広さではないがエージェントの作業量が均一となるべきである。これによって、エージェント間の公平性を実現し、効率的に領域を清掃できる。

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の推定値の逆数を、隣接した領域の担当エージェントと比較し、大きい方が一部の隣接領域に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] の手法では、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よって不要に分断される場合がある。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。

¹ 早稲田大学
Waseda University

他領域を通過することは、特定の場所への過剰な訪問などの無駄な重複や、担当領域の大きさを越えた範囲を巡回することになるなど、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実際よりゴミの量を少なく見積もって行動を変えろという、誤った行動を発生させる。しかし、環境の形状によっては領域の分断が必要なこともある。そのため、本研究では孤立した担当領域を減らし、不要な他領域の通過を抑えた領域分割配置の実現を目的とする。他領域の通過が必須となる領域を含み、かつ領域の分断が発生しやすい特定の環境を例題とした実験を行い、提案手法の評価をする。また、分断の発生状況を図示し、実際に提案手法が不要な分断を防ぐことを示す。

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、[2]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。[3]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。[4]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。こ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。[5]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。

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。[6]は[2]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り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。[7]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、[6]、[7]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、[7]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は考慮されていない。[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ることを考慮していない。また、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。[8]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。[9]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。本研究では[1]を拡張し、不要な領域の分断を減少させ、効率化を目指している。

3. モデルの定義

この節では[1]をベースに本研究で使用する環境とエージェントのモデルを説明し、[1]で提案された領域分割手法を説明する。

3.1 環境のモデル

$A = \{a_1, \dots, a_N\}$ をエージェントの集合とする。時間は離散時間とし、最小単位はステップと呼ぶ。1ステップの間にエージェントは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 $V = \{v_1, \dots, v_x\}$ はノードの集合であり、 E はエッジの集合である。ノード $v_i, v_j \in V$ 間にエッジが存在する場合、そのエッジを $e_{i,j}$ と表す。本研究では、必要であればダミーのノードを加え、全エッジの重みは1とし、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と1ステップで移動可能とする。エージェント a_i は定められた位置に充電基地 $v_{base}^i (\in V)$ があり、そこで充電を行う。

ある時刻 t にエージェント a_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。環境全体で均一に清掃を行うために、エージェント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 ($0 \leq p_v \leq 1$) でゴミが発生するものとする。ノード v の時刻 t における蓄積したゴミの量 $L_t(v)$ は、

$$L_{t+1}(v) = \begin{cases} L_t(v) + 1 & (p_v) \\ L_t(v) & (1 - p_v) \end{cases} \quad (1)$$

と表される。エージェントが時刻 t にノード v へと到達したとき、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には、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することである。現在多くのマップ作成のアルゴリズム [10][11] 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、将来的には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はエージェントが同一のノードに複数存在することができ、その際の衝突は考えない。現実では複数台のエージェントが同一の場所に存在はできず、同一地点に移動しようとするとう衝突が発生する。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する。第3は各エージェントは有限容量のバッテリーを保持することである。そのためエージェントは探索を無限に続けることは不可能であり、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

エージェントには探索状態と充電状態という2つの状態がある。探索状態とはエージェント a_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、自身の担当領域の清掃を行う。ここで a_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 a_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、エージェントは状況によって到達できないノードが存在する。エージェント a_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、

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

と表される。そのため

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

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 へは移動せず、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 a_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。ここでは充電にかかる時間はバッテリー消費量に比例すると仮定し、その充電にかかる時間は比例定数 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。移動開始の際にバッテリーが最大まで充電することは必須ではない。しかし、本研究においては簡単のため、エージェントは最大まで充電してから移動を開始する。

3.4 初期担当領域

エージェント a_i は充電基地 v_{base}^i から距離 d_{init} 以下の範囲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 a_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 G_{enc}^i を飛び地の集合とする。明らかに、連結領域は互いに素である。飛び地の集合 G_{enc}^i の大きさ $|G_{enc}^i|$ を担当領域の飛び地の数と呼ぶ。なお、 $|G_{enc}^i| \geq 1$ 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 a_i の担当領域 G_t^i について、 G_t^i 全てを限なく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。通過領域の決定のための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き優先探索 (DDFE) とし、本実験中に他の探索アルゴリズムは用いない。以下でDDFEのアルゴリズムを説明する。

エージェント a_i は充電基地を出発する際に、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選択して、ターゲットノードと設定し最短経路で移動する。ターゲットノードへと到達したら、隣接するノードの中でまだ訪れていないノードをランダムに1つ選択して移動し、そのノードをスタックの上部にプッシュする。エージェント a_i が訪れたことのない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 a_i の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 a_i は再び訪れていないノードを選択する。この動作を繰り返した後に、最初に選択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、自らの担当領域を拡大しようとする試みである。エージェントが領域拡大行動を試みるのは次に示す条件を満たしたときである。エージェント a_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 γ 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 $t_b + \gamma$ 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、実際の状況とは大きく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 a_i は時刻 t_b に開始した巡回中に、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に、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。

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 a_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 a_i は自身の担当領域の隣接ノードの内、自身の担当領域内のノードの集合 V_t^i 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 a_i は B の中から、 I_{avoid}^i (これは後に説明する) に含まれず、かつ自分の基地 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 a_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 a_i は時刻 t に充電基地へ帰還した際に、その時刻での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 a_i

の拡大力 $\varepsilon(i, t)$ は $E(L(V_t^i))$ を用いて

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

とする。ただし $E(L(V_t^i)) = 0$ の場合 $\varepsilon(i, t)$ は十分に大きな整数値とする。各エージェントは再び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 a_i は I^i として定めた拡大対象のノードに対して、そのノードを現在担当領域に持っているエージェント a_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 a_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 a_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に変化はない。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。

$$D_{t_s, t_e}(V) = \sum_{t=t_s}^{t_e} L_t(V) / (t_e - t_s + 1) \quad (8)$$

さらに、各エージェントの作業量が均一であるかを確認するために、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

今回提案する手法は [1] 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し、より効果的な領域分割を行う手法である。

4.1 飛び地発生の緩和

提案手法において、エージェント a_i は飛び地の発生を緩和するために、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に譲渡する。 a_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 $G_{enc_j}^i$ に対してそれぞれ次の評価値 E_{enc_j} を計算する。ここで $Pot^i(v)$ はノード v におけるエージェント a_i の充電基地

v_{base}^i からの距離である。なお、距離は第 3.4 節で定義したものをを用いる。

$$E_{enc_j}^i = \frac{1}{|G_{enc_j}^i|} \sqrt{\sum_{v \in G_{enc_j}^i} Pot^i(v)^2} \quad (9)$$

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

$$E_{enc_j}^i > \frac{|G_{enc_j}^i|}{R_3 \cdot |G_{enc_j}^i|} \quad (10)$$

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、その中の 1 つをランダムに選択し譲渡の対象とする。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。

4.2 通過領域の決定と担当領域の巡回

エージェント a_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、それぞれの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} を次の通り定める。ただし G_{ins} は初期値として $G_{ins} = \phi$ とおく。

- (1) a_i はノードの集合 G_{ins} 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} 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 $G_{enc_n}^i$ とする。
- (3) G_{ins} に、 $G_{enc_n}^i$ に含まれるノード、及び G_{ins} と $G_{enc_n}^i$ 間の最短経路上のノードを加える。
- (4) G_{ins} に含まれてい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} から a_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 a_i の通過領域とする。

エージェント a_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

提案手法の性能と特徴を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 に示すように 10×10 の 2 次元のグリッドグラフと 20×20 の 2 次元のグリッドグラフが、1 つのエッジで連結しているグリッドグラフとする。各エージェント $A = \{a_1, a_2\}$ は充電基地 v_{base}^i ($i = 1, 2$) を出発し G を巡回清掃する。また G の各ノード v におけるゴミの発生確率は一様に $p_h = 2 \times 10^{-4}$

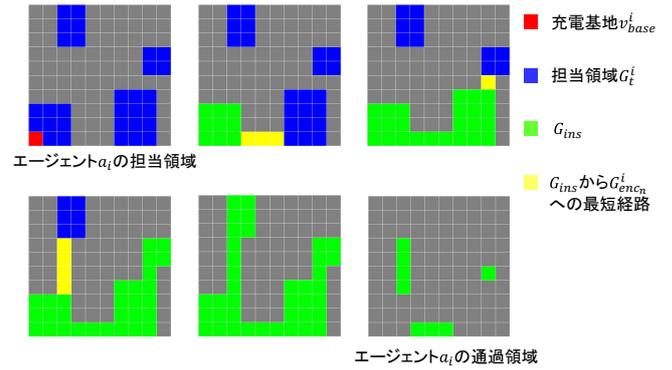


図 1: 通過領域決定アルゴリズム

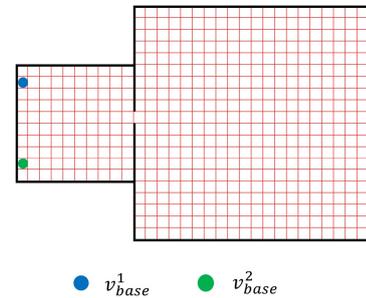


図 2: エージェントが清掃を行う空間 G

表 1: 実験設定

	パラメータ	値
エージェント台数	$ A $	2 台
拡大戦略を行うためのパラメータ	R_1	0.7
	R_2	0.7
	R_3	0.01
	γ	300
拡大戦略の制御パラメータ	k_{inc}	15
	k_{avoid}	30
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

とした。

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。

実験は 1 回 100 万ステップで行い、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) 各担当領域の広さの推移
- (3) 各担当領域のゴミ残量の総和の推移
- (4) 各担当領域の飛び地の数の推移
- (5) 全エージェントの通過領域の大きさの和の推移

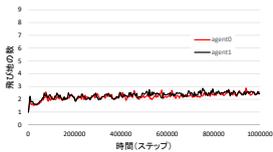


図 3: 各担当領域の飛び地の数の推移 [提案手法]

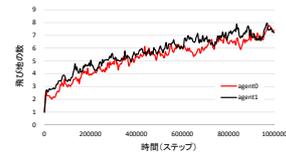


図 4: 各担当領域の飛び地の数の推移 [既存手法]

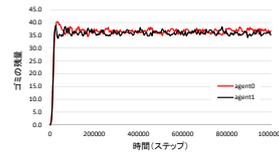


図 9: 各担当領域のゴミ残量の推移 [提案手法]

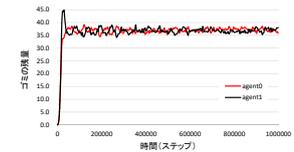


図 10: 各担当領域のゴミ残量の推移 [既存手法]

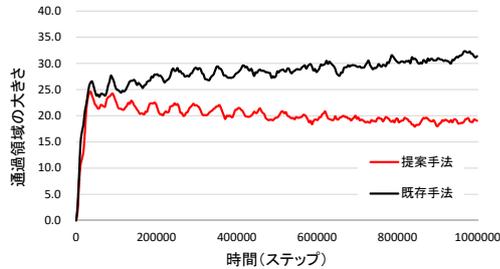


図 5: 全エージェントの通過領域の大きさの和の推移

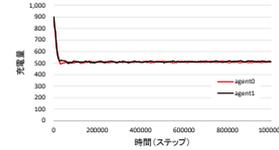


図 11: 各エージェントの 3600 ステップの間の最小充電量の推移 [提案手法]

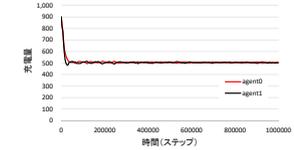


図 12: 各エージェントの 3600 ステップの間の最小充電量の推移 [既存手法]

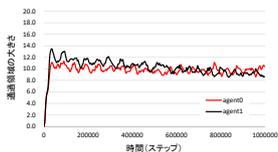


図 6: 各エージェントの通過領域の大きさの推移 [提案手法]

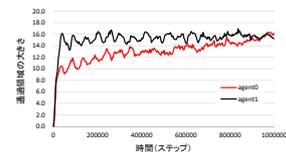


図 7: 各エージェントの通過領域の大きさの推移 [既存手法]

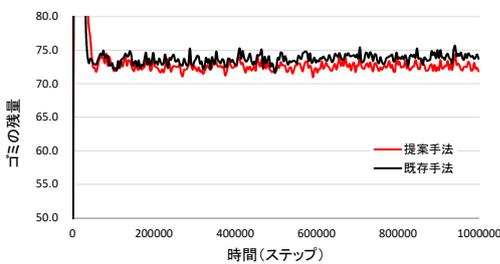


図 8: 全ノードにおけるゴミ残量の総和の推移

- (6) 各エージェントの通過領域の大きさの推移
- (7) 各エージェントの 3600 ステップの間の最小充電量の推移

について調査する。実験結果のデータは上記の実験を 100 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過領域の通行を加えたものを採用した。

5.2 実験結果

各担当領域の飛び地の数の推移を図 3, 図 4 に示す。100 万ステップ経過時に、提案手法では両方のエージェントの飛び地の数が約 2.5 個にほぼ収束しているのに対し、既存

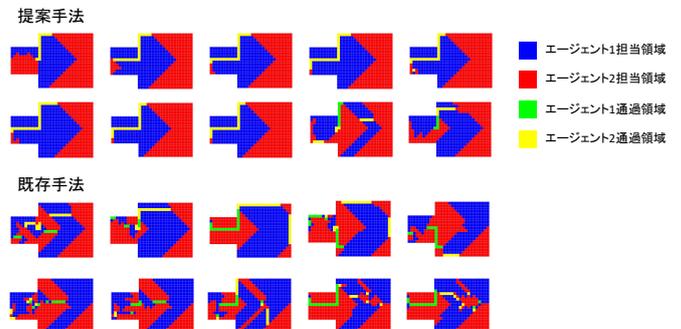


図 13: それぞれの手法における領域分割の例

手法ではどちらのエージェントも 6 個を超えて増加を続けている。また、それぞれの手法における全エージェントの通過領域の大きさの総和の推移を図 5 に、各エージェントの通過領域の大きさの推移を図 6, 図 7 に示す。これらの図より、約 40 万ステップが経過した後は、エージェント全体、あるエージェント単体のどちらの場合も、提案手法の方が通過領域の発生を抑えたことがわかる。実際のシミュレーションにおける領域分割の例を図 13 に示す。

次に全ノードのゴミ残量の総和の推移を図 8 に示す。この図より、提案手法の全ノードにおけるゴミ残量の総和は既存手法に比べ約 2% 程度減少したことがわかる。次に、各担当領域のゴミの残量の推移を図 9, 図 10 に示す。これらの図より既存手法、提案手法ともにそれぞれのエージェントの担当領域のゴミの残量がほぼ等しくなっている。

最後に各エージェントの 3600 ステップの間の最小充電量の推移を図 11, 図 12 に示す。これらの図より各手法でエージェントは最大でも 400 ほどしか充電を消費していないことがわかる。

5.3 考察

図3, 図4, 図5, 図6, 図7から飛び地の数と通過領域の大きさは, 提案手法の方が既存手法に比べて小さいことがわかる. これは, エージェント間で飛び地の譲渡を行うというプロセスが, 本研究の目的である領域分割配置の実現に対して一定の効果があることがわかる. 具体的な領域分割の例は図13に示した通りで, 提案手法の飛び地の数, 通過領域の大きさが既存手法と比べ減少したことがわかる.

図9, 図10より, 各担当領域のゴミの残量がほぼ均一になっているため, どちらの手法でも各エージェントのゴミ回収量が均一となる領域分割が実現できたと言える. しかし, 図8が示す清掃を行う空間全体でのゴミの残量は, 提案手法の方が約2%少ない. 図5, 図6, 図7に示された結果から, この差は第1節で言及した, 他領域の通過による作業の重複によって引き起こされたと推測できる. 通過領域に含まれるノードに対するそれぞれのエージェントの過度な訪問と, 通過領域を含めて巡回を行うことによる自身の担当領域のノードへの訪問間隔の増加が, 図8で示した空間全体でのゴミの残量の差が生じる原因と考えられる.

図8に示されている通り, 既存手法と比べ提案手法の全ノードのゴミの残量の総和は小さかった. しかし, その減少量は約2%程度に留まっている. これは環境の小ささとエージェントの台数の少なさに起因すると考えられる. まず環境の小ささについて議論する. 図11, 図12から, 各手法でエージェントは最大でも400程度しか充電を消費していない. これは本実験ではほとんどの場合において, 1度の巡回でエージェントは充電基地に戻ることなく, 自らの担当領域内全てのノードを訪れられることを示している. 表1よりエージェントは, 清掃空間を巡回する時間に対して3倍の充電時間を必要とする. そのためあるエージェントについて, 1度の巡回で訪問できないノードが存在する状態で充電状態に入るとき, それらのノードの訪問間隔が毎回の訪問ができるときのノードの訪問間隔と比べ, かなり大きくなる. 反対に1度の巡回で担当領域内全てのノードを訪問できている場合は, 巡回するノード数に多少の差があっても効率はあまり変わらない.

次にエージェントの台数の少なさについて考える. エージェントが他領域を通過した際にそのノードのゴミを回収することによって, そのノードを担当領域に持つエージェントは式7に示される条件を満たしにくくなる. これにより, 本来想定されているよりも領域拡大行動が発生しづらくなるという現象が発生し, 領域分割に偏りが生じる. この現象は通過領域が大きくなればなるほどその影響も大きくなる. しかし, 今回の実験ではエージェントは2台であるため各エージェントの通過領域はそれぞれ互いの担当領域である. そのためお互いの領域拡大行動を抑制し合い, 他領域を通過したことによる影響が相殺された状態になったと考えられる. これらの理由により, 図5に示され

た通過領域の大きさの差による影響が抑えられ, 全体の清掃効率には大きな違いが現れなかったと考えられる.

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, 領域拡大の際に発生する飛び地の増加を抑制し, より効果的な領域分割配置を実現する手法を提案した. 巡回清掃では清掃空間の形状やエージェントの充電基地の配置によって, 各担当領域の飛び地の数が増加するときがある. それによりエージェントの通過領域の大きさが増大して, 各担当領域内のノードの訪問間隔の増加や適切な領域拡大行動の妨げが起ころう. そのため本研究では各エージェントの領域拡大行動によって発生した飛び地について, 条件を満たした飛び地を他のエージェントに譲渡することで, 飛び地の数と通過領域の大きさを抑えた. 評価実験の結果, 飛び地の数と通過領域の大きさを減少させて, 提案手法の有効性を示すことができた.

今後の課題として, 環境をより複雑化することが挙げられる. 本論文ではエージェントは2台, 環境の構造は大きさの異なる2つの正方形格子状2次元グリッドグラフを繋げたもの, 全体のノード数も500と非常に簡易で小さな環境での実験となった. またゴミの発生確率は環境全体で均一で, その値は既知とした. 第5.3節でも述べた通り, 環境が簡易なものであると環境全体のゴミの残量の総和の差が結果に表れにくくなる. 現実の環境はいくつもの構造が複合的に存在するため複雑であり, ゴミの発生確率も不均一で未知なことがほとんどである. よってエージェント台数の増加, 環境構造の複雑化, 環境の大きさの拡大, ゴミの発生確率の不均一化や未知化など, 現実にも似た条件下で実験結果がどのように変化するのか調査する必要がある.

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] 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).
- [3] 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).
- [4] 米田圭佑, 加藤千紘, 菅原俊治: バッテリ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).
- [5] K., Y., C., K. and T., S.: 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).
- [6] 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).
 - [7] 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).
 - [8] Nasir, A., Salam, Y. and Saleem, D. Y.: Multi-Level Decision Making in Hierarchical Multi-agent Robotic Search Teams, Vol. 1 (2009).
 - [9] 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).
 - [10] 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).
 - [11] Wolf, D. F. and Sukhatme, G.: Mobile Robot Simultaneous Localization and Mapping in Dynamic Environments, *Autonomous Robots 19(1)*, pp. 53–65 (2005).