# SCI'17論文 特集号—I

論 文

# 通信可能範囲を考慮した 分散協調制御システムに基づく迷路探索\*

望月優加理†・澤田 賢治‡・新 誠一†

Maze Exploration Algorithm via Distributed Collaborative Control System Considering Communicable Range\*

Yukari Mochizuki<sup>†</sup>, Kenji Sawada<sup>‡</sup> and Seiichi Shin<sup>†</sup>

When multi-agent systems explore in actual environments, it is difficult to manage agents centralizedly from outside due to communicable range. This paper proposes a maze exploration algorithm for distributed control system considering communicable range of agents. In proposed algorithm, agents share information each other within communicable range and then each agent detects deadlocks generating a wait-for graph from the shared information. Sharing cycle of wait-for graphs enables agents to take over the explorations each other and resolve deadlocks. This paper verifies the effectiveness of the proposed deadlock resolution algorithm via numerical experiments about some of communicable ranges and numbers of agents of exploration system. The simulation results show the influence of the proposed deadlock resolution algorithm on the efficiency of exploration and the path length of agents. The simulation results also show the relationship between the communicable range, the number of agents of exploration system and the failure probability of exploration.

#### 1. はじめに

マルチエージェントシステムの迷路探索はさまざまな分野で研究されてきた横断的研究課題の一つである. 計算機科学の分野ではMARTA\*[1]をはじめとする,マルチエージェントシステムを用いたグラフ探索アルゴリズム [2,3]が研究されている.本論文ではエージェント間での衝突回避を考慮した,異なる役割をもつエージェント群による迷路の協調探索に取り組む.文献[3]では異なる役割をもつエージェント群による協調探索手法が提案されているが,エージェント間での衝突回避は考慮されていない.文献[4]が提案する協調探索手法は,経路

避は移動経路の変更の原因となり、エージェントが役割に応じて与えられた目標を達成するまでの移動経路長が増加する恐れがある。移動経路の変更を要さないデッドロックの解決手法として、エージェントの移動経路を利用したデッドロックの予測とエージェントの移動順序の変更による回避で構成されるデッドロック解決手法[5]が挙げられる。文献[6]によると移動順序の変更を用いたデッドロック回避では、再経路計画によるデッドロック回避[7]や退避行動を含むデッドロック検知・回復[8]と比較して移動経路の変化が抑えられる。しかしながら、デッドロック発生の予測は探索終了までの移動経路の情報を必要とするため、長期の経路計画が困難な迷路など

計画の段階でエージェント間の衝突・デッドロック回避 を考慮する。ただし、経路計画段階でのデッドロック回

著者らは異なる役割をもつエージェント群による迷路の協調探索におけるデッドロック検知・回復手法 [9] を提案している。文献 [9] のデッドロック検知手法は Wait-for graph [10] を利用したデッドロック予測手法 [5] を未知環境での探索問題におけるデッドロック検知に拡張したものである。また、デッドロック回復動作において役割交換と同時に探索引継ぎを行うことで、デッドロック解消のためのエージェントの移動経路の変化を抑えている。

の未知環境下における探索問題への適用は困難である.

Key Words: multi-agent system, maze exploration, deadlock resolution.

<sup>\*</sup> 原稿受付 2017年7月31日 第61回システム制御情報学会研究発表講演会にて発表 (2017年5月)

<sup>&</sup>lt;sup>†</sup> 電気通信大学 大学院 情報理工学研究科 Graduate School of Informatics and Engineering, The University of Electro-Cmmunications; 1-5-1, Chofugaoka, Chofu city, Tokyo 182-8585, JAPAN

<sup>‡</sup> 電気通信大学 i-パワードエネルギー・システム研究センター Info-Powered Energy System Research Center, The University of Electro-Cmmunications; 1-5-1, Chofugaoka, Chofu city, Tokyo 182-8585, JAPAN

文献 [9] は複数のエージェントとエージェントの動きを 統制するサーバで構成される集中管理型の探索システム を対象としている。しかしながら、実環境下での迷路探 索ではエージェントの通信範囲の制約により外部からの エージェントへの指示が困難となる状況が想定される。

そこで、文献 [9] で提案した協調探索手法をエージェントの通信可能範囲に制限をもつ分散制御型探索システムによる協調探索問題に拡張する。本論文では文献 [9] で提案したデッドロック検知・回復手法を拡張し、限られた通信範囲内での情報共有によるデッドロック検知と役割交換、探索引継ぎによるデッドロック解消の実現を目指す。提案する協調探索手法ではエージェント間での探索引継ぎと役割交換を表現するために、探索ユニットと疑似移動を取り入れている。探索ユニットはエージェントの探索記録と役割を、疑似移動はエージェント間での探索引継ぎ動作を表す。

はじめに、探索ユニットへの探索記録、疑似移動による探索引継ぎおよび経路計画手法の使い分けに対応する、エージェントの基本機能の数理モデルを示す. 基本機能は迷路探索においてエージェントが単体で扱う機能であり、周囲のエージェントとの情報共有を必要としない. つぎに、情報共有手段と迷路の地図情報の共有機能を作成し、その数理モデルを示す. 作成した情報共有手段をもとに、デッドロックの検知・回復手法の詳細とその数理モデルを示す. 数理モデルをもとに通信可能範囲と検知が保証されるデッドロックの規模の関係を考察する. 提案するデッドロック解決手法は、各エージェントの通信可能範囲内で発生したデッドロックの解消を可能にする.

数値実験により、提案手法によるデッドロック回復機能の動作検証を行う、また、迷路探索の効率、エージェントの稼働率の面から提案するデッドロック検知・回復手法の特徴とエージェントの台数、通信可能範囲が迷路探索における目標達成の成否に与える影響を考察する。

#### 2. 問題設定

本論文では移動,自己位置同定,センサによる障害物と移動物体の検知,自律的な移動経路の決定が可能な自 律移動ロボット群による迷路の協調探索について考える.

## 2.1 探索対象

探索対象は区画と隣接する区画間の境界で構成される空間である。本論文では探索対象を有向グラフ M=(V,E) で表現する。V,E はそれぞれ区画を表すノードと,隣接する 2 区画の境界を表すアークの有限集合である。 $v_x \in V$  は x 番目 (x=1,2,...,|V|) のノードを表し, $e_{xy}=(v_x,v_y)\in E$  は  $v_x$  から  $v_y$  へつながるアークを表す。 $e_{xy}$  には固有の番号  $\gamma(x,y)\in \{1,2,...,|E|\}$  が与えられる。以降では, $v_x$  から  $v_y$  への移動を  $e_{xy}$  の通過と表す.探索対象の例として 2 次元迷路とそのグラフ表現を Fig. 1

に示す. Fig. 1(a) は隣接する区画間の境界を点線で,境 界上の障害物を実線で示す.

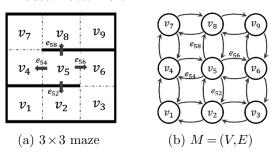


Fig. 1 Exploration field

Fig. 1(b) に示すように M には探索対象の探索範囲や 区画間の隣接関係のみが反映され、(a) 上の壁の位置情報を含まない。探索対象の形状は M とアーク上の障害物の有無によって定義される。

ノードとアークの接続関係は行列  $B_+ \in \{0,1\}^{|V| \times |E|}$  と  $B_- \in \{0,1\}^{|V| \times |E|}$  で表す。  $B_+$  と  $B_-$  の x 行  $\gamma(y,y')$  列要素をそれぞれ  $b_+{}^x_{y,y'}$ ,  $b_-{}^x_{y,y'}$  と表す。  $v_x$  が  $e_{yy'} \in E$  の終点である場合 (x=y'),  $b_+{}^x_{y,y'}$  は 1 となる。 そうでない場合, $b_+{}^x_{y,y'}$  は 0 となる。  $v_x$  が  $e_{yy'} \in E$  の始点である場合 (x=y),  $b_-{}^x_{y,y'}$  は 1 となる。 そうでない場合, $b_-{}^x_{y,y'}$  は 0 となる。  $B_+$  と  $B_-$  はアークに関する情報を,それぞれアークの終点と始点に関する情報に変換する。アーク上の障害物の有無をベクトル  $w \in \{0,1\}^{|E|}$  と表す。 w の  $\gamma(x,y)$  番目の要素を  $w_{x,y}$  と表す。  $e_{xy} \in E$  上に障害物が存在する場合, $w_{x,y}$  は 1 となる。そうでない場合, $w_{x,y}$  は 0 となる。

#### 2.2 探索システム

対象とする探索システムは自律移動ロボットを模した 複数のエージェントとエージェントと同数の探索ユニットで構成される分散制御型探索システムである.探索ユニットはエージェントの役割を表し,エージェントの探索記録をもつ.本論文ではエージェントの役割を探索ユニットがもつミッションと経路計画手法の組合せで定義する.ミッションは目的地への到達や迷路形状の把握など複数の種類が存在し,各探索ユニットに一つずつミッションが割り当てられる.探索記録は迷路の地図情報と各時刻におけるエージェントの状態および移動経路で構成される.探索ユニットは各エージェントに一つずつ割り当てられ,エージェント間での引き受け,引き渡しによる再割り当てが可能である.エージェントは探索ユニットが示す役割に従って個別に行動を決定し,行動の結果を探索記録として探索ユニットに記録する.

ここで,通信可能範囲に制限をもつI台のエージェントとJ個 ( $J \le I$ ) のミッションをもつ探索システムによる分散協調探索について考える.本論文では通信可能範囲内での情報共有による,迷路情報の共有,デッドロック解決の実現を目指す.デッドロック解決ではエージェントの移動経路の変更を抑えるために,エージェント間

での探索引継ぎと役割交換を用いたデッドロック検知・回復に取り組む. 以降では探索システムにおけるi番目 (i=1,2,...,I) のエージェントをエージェントiと表記し、このときのiの値をエージェントの ID として扱う. 同様に、探索システムにおけるi番目 (i=1,2,...,I) の探索ユニットをユニットiと表記し、このときのiの値を探索ユニットの ID として扱う.

# 3. エージェントの基本機能

本章では探索ユニットの再割り当てによる役割交換とそれに伴う経路計画手法の使い分け、エージェント間での探索引継ぎを可能にする迷路探索の基本機能について述べる。基本機能はエージェントの自己位置・状態同定、障害物検知、経路計画、移動、他のエージェントとの衝突回避、探索行動記録で構成される。本論文ではエージェントの基本機能に以下の制約を設ける。

- a) 全エージェントは区画移動と周辺情報の取得を繰り返し行い、移動のタイミングは同期される.
- b) 一つのノードに複数のエージェントが同時に移動 することはできない. また, 他のエージェントが 存在するノードへの移動はできない.
- c) 1回の区画移動における移動距離は最大で1区画である.
- d) エージェントは自身が存在するノードに接続する アークの状態を検知可能である.
- e) エージェントは周辺のエージェントの位置や速度 を測定することで、同一のノードへの移動を試み る他のエージェントを検知可能である.
- f) 探索中のエージェントの状態は初期状態と探索記録によって一意に求まる.

## 3.1 自己位置・状態同定

エージェントは直前の移動記録をもとに自身の現在位置と状態の同定を行う. 状態はエージェントの向きなどによって定義され. 位置は含まない.

時刻 t におけるエージェント i の状態を  $S^i(t)$  と表す. デッドロックの発生に直接関係しないため,本論文では  $S^i(t)$  の詳細は省略する.制約 f) より  $S^i(t)$  の各要素は時刻 t においてエージェント i に割り当てられたユニットに記録されている移動記録にしたがって変化するものとする.時刻 t におけるエージェント i の位置を  $z^i(t) \in \{0,1\}^{|V|}$  と表す. $z^i(t)$  の x 番目の要素を  $z^i_x(t)$  と表す.時刻 t においてエージェント i が  $v_x$  に存在する場合, $z^i_x(t)$  は 1 となる.そうでない場合, $z^i_x(t)$  は 0 となる.時刻 t におけるエージェント i の移動経路が  $u^i(t) \in \{0,1\}^{|E|}$  と与えられたとき, $z^i(t+1)$  は

$$z^{i}(t+1) = z^{i}(t) + (B_{+} - B_{-})u^{i}(t)$$
 (1)

と求まる.  $z^i(t)$  は

$$\boldsymbol{z}^{i}(t)^{\mathrm{T}}\boldsymbol{1}_{|V|} = 1 \tag{2}$$

を満たす.ここで, $\mathbf{1}_n$  はすべての要素が1であるn個の要素で構成される列ベクトルである.

#### 3.2 障害物検知

迷路形状の把握と障害物回避のためにエージェントは 周辺のアーク上の障害物の有無を検知する. 制約 c), d) により、エージェントの障害物回避が可能となる、時刻 tにおけるエージェントiの障害物検知の結果と検知範 囲をそれぞれ  $\boldsymbol{a}^{i}(t) \in \{0,1\}^{|E|}, \ \boldsymbol{l}^{i}(t) \in \{0,1\}^{|E|}$  と表す.  $\mathbf{a}^{i}(t)$  と  $\mathbf{l}^{i}(t)$  の  $\gamma(x,y)$  番目の要素をそれぞれ  $a_{xy}^{i}(t)$ ,  $l_{x,y}^{i}(t)$  と表す. 時刻 t においてエージェント i が  $e_{xy} \in E$ 上に障害物を検知した場合,  $a_{x,y}^i(t)$  は1となる. そうで ない場合、 $a_{xy}^i(t)$  は 0 となる。時刻 t において  $e_{xy} \in E$ がエージェントiの検知範囲に含まれるとき、 $l_{x,y}^{i}(t)$ は 1となる. そうでない場合,  $l_{x,y}^i(t)$  は 0 となる. 制約 d) は、 $\boldsymbol{l}^{i}(t)$  を用いて $\boldsymbol{l}^{i}(t) \geq (\boldsymbol{B}_{+} + \boldsymbol{B}_{-})^{\mathrm{T}} \boldsymbol{z}^{i}(t)$  と表される. エージェントは検知結果をもとに、各アークの状態 を"未知"と"既知"の二つに分類する. 状態"未知"は アーク上の障害物の有無が未知であることを表し、状態 "既知"はアーク上の障害物の有無が既知であることを 表す. さらに. 状態 "既知" のアークを "あり" と "なし" の二つの状態に分類する. 状態 "あり" はアーク上に障 害物が存在することを表し、状態"なし"はアーク上に 障害物が存在しないことを表す. 以降では状態 "あり" か否かの判定を迷路の形状情報、状態"未知"か"既知" かの判定を検知状況情報と表す。また、迷路の形状情報 と検知状況情報を併せて地図情報として扱う。エージェ ントは得られた迷路の地図情報を探索ユニットに記録す る. 時刻tにおいてエージェントiに割り当てられる探 索ユニットの ID を  $\mu^i(t) \in \{1,2,...,I\}$  と表す。時刻 t に おけるユニットiの迷路の形状情報と検知状況情報をそ れぞれ $\hat{\boldsymbol{w}}^{i}(t) \in \{0,1\}^{|E|}, \hat{\boldsymbol{q}}^{i}(t) \in \{0,1\}^{|E|}$ と表す.  $\hat{\boldsymbol{w}}^{i}(t)$ と $\hat{m{q}}^{i(t)}$ の $\gamma(x,y)$ 番目の要素をそれぞれ $\hat{w}_{x,y}^{\hat{i}}(t)$ ,  $\hat{q}_{x,y}^{\hat{i}}(t)$ と表す. 時刻tにおけるユニットiの地図情報において  $e_{xy} \in E$  が状態 "あり" に分類される場合,  $\hat{w}_{x,y}^i(t)$  は 1 となる. そうでない場合,  $\hat{w}_{x,y}^i(t)$  は 0 となる. 時刻 t に おけるユニットiの地図情報において $e_{xy} \in E$ が状態"既 知"に分類される場合、 $\dot{q}_{x,y}^i(t)$ は1となる。そうでない 場合、 $\dot{q}_{x,y}^i(t)$  は 0 となる.

エージェント i は  $\hat{\boldsymbol{w}}^{\mu^i(t)}(t-1)$  と  $\boldsymbol{a}^i(t)$  の重ね合わせを新たな迷路の形状情報  $\tilde{\boldsymbol{w}}^i \in \{0,1\}^{|E|}$  として求める.同様に, $\hat{\boldsymbol{q}}^{\mu^i(t)}(t-1)$  と  $\boldsymbol{l}^i(t)$  の重ね合わせを新たな検知状況情報  $\boldsymbol{q}^i(t) \in \{0,1\}^{|E|}$  として求める. $\tilde{\boldsymbol{w}}^i(t)$  と  $\boldsymbol{q}^i(t)$  はそれぞれ

$$\tilde{\boldsymbol{w}}^{i}(t) = (\boldsymbol{I} - \boldsymbol{L}^{i}(t)) \hat{\boldsymbol{w}}^{\mu^{i}(t)}(t-1) + \boldsymbol{a}^{i}(t)$$
(3)

$$\mathbf{q}^{i}(t) = (\mathbf{I} - \mathbf{L}^{i}(t)) \, \dot{\mathbf{q}}^{\mu^{i}(t)}(t-1) + \mathbf{l}^{i}(t) \tag{4}$$

と求まる.  $L^i(t) \in \{0,1\}^{|E| \times |E|}$  は  $\gamma(x,y)$  番目の要素に  $l^i_{x,y}(t)$  をもつ対角行列であり,アークに関する情報から

時刻 t におけるエージェント i の検知範囲の情報を抽出する。エージェントは  $\tilde{w}^i(t)$  と  $q^i(t)$  を共有し、共有情報をもとに地図情報を更新する。情報共有を行わない場合の  $\hat{w}^{\mu^i(t)}(t)$ ,  $\dot{q}^{\mu^i(t)}(t)$  はそれぞれ  $\tilde{w}^i(t)$ ,  $q^i(t)$  と一致する。

#### 3.3 経路計画

経路計画ではエージェントの状態をもとに、探索ユニットが指定する経路計画手法に従ってエージェントの移動経路を決定する。時刻 t におけるエージェントi の経路計画結果を $\hat{\boldsymbol{u}}^i(t) \in \{0,1\}^{|E|}$  と表す。 $\hat{\boldsymbol{u}}^i(t)$  の $\gamma(x,y)$  番目の要素を $\hat{u}^i_{x,y}(t)$  と表す。時刻 t におけるエージェントi の経路計画が $e_{xy} \in E$  の通過を表す場合, $\hat{u}^i_{x,y}(t)$  は1となる。そうでない場合, $\hat{u}^i_{x,y}(t)$  は0となる。制約 $\mathbf{c}$ ) は $\hat{\boldsymbol{u}}^i(t)$  を用いて $\hat{\boldsymbol{u}}^i(t)^{\mathrm{T}}\mathbf{1}_{|E|} \leq 1$  と表される。

#### 3.4 移動

このとき, 制約b)に従った衝突回避が同時に行われ る. 本論文では制約 e) に定義される機能を用いた衝突回 避を、占有区域にもとづく他のエージェントとの衝突リ スクの検知によって表現する. エージェントは他のエー ジェントとの衝突リスクを検知した場合に停止行動を選 択する. 占有区域をエージェントの存在位置と移動経路 の終点位置の集合とおくと、制約 b) が示す状況は占有 区域の重なりで表される. 時刻tにおけるエージェントiの占有区域を  $o^i(t) \in \{0,1\}^{|V|}$  と表す.  $o^i(t)$  の x 番目の 要素を $o_x^i(t)$ と表す. 時刻tにおいて $v_x$ がエージェントiの占有区域に含まれる場合、 $o_x^i(t)$  は1となる。そうでな い場合,  $o_x^i(t)$  は 0 となる. 時刻 t におけるエージェント iのエージェントjに対する衝突リスクを $p_i^i(t) \in \{0,1\}$ と表す. 時刻tにおいてエージェントiがエージェントjとの衝突リスクを検知した場合,  $p_i^i(t)$  は1となる. そう でない場合,  $p_i^i(t)$  は 0 となる.  $o^i(t)$  は

$$\boldsymbol{o}^{i}(t) = \boldsymbol{z}^{i}(t) + \boldsymbol{B}_{+}\boldsymbol{u}^{i}(t) \tag{5}$$

と与えられる. このとき, 制約 b) は  $o^i(t)$  を用いて

$$\sum_{i=1}^{I} \boldsymbol{o}^{i}(t) \leq \mathbf{1}_{|V|} \tag{6}$$

と表せる。複数のエージェントが同一のノードへの移動を試みた場合,より ID の小さなエージェントの移動を優先する。時刻 t におけるエージェント i から見たエージェント j の占有区域を  $\check{o}^i(t,j) \in \{0,1\}^{|V|}$  と表す。 $\check{o}^i(t,j)$  は優先順位を考慮して

$$\check{o}^{i}(t,j) = \begin{cases} z^{j}(t), & \text{if} \quad j \geq i \\ o^{j}(t), & \text{if} \quad j < i \end{cases}$$
 (7)

と与えられる. このとき, (5), (7) 式より  $\check{o}^i(t,j)$  は

$$\mathbf{z}^{j}(t) \leq \check{\mathbf{o}}^{i}(t,j) \leq \mathbf{o}^{j}(t)$$
 (8)

を満たす. (8) 式より, (6) 式が満たされるとき

$$\sum_{j=1}^{I} \mathbf{z}^{j}(t) \leq \sum_{j=1}^{I} \check{o}^{i}(t,j) \leq \mathbf{1}_{|V|}$$
 (9)

が成立する。本論文では $\hat{u}^i(t)$ の終点が $\check{o}^i(t,j)$ が示す占有区域に含まれる状況をエージェントiのエージェントjに対する衝突のリスクとして扱う。このとき、 $p_i^i(t)$ は

$$p_i^i(t) = \check{\boldsymbol{o}}^i(t,j)^{\mathrm{T}} \boldsymbol{B}_+ \hat{\boldsymbol{u}}^i(t)$$
 (10)

と求められる. (6) 式を満たす  $u^i(t)$  として

$$\mathbf{u}^{i}(t) = \begin{cases} \hat{\mathbf{u}}^{i}(t), & \text{if } \sum_{j=1}^{I} p_{j}^{i}(t) = 0\\ \mathbf{0}, & \text{if } \sum_{j=1}^{I} p_{j}^{i}(t) = 1 \end{cases}$$
(11)

を用いる.  $(9)\sim(11)$  式より  $\boldsymbol{u}^{i}(t)$  は

$$\sum_{i=1}^{I} \check{\boldsymbol{o}}^{i}(t,j) + \boldsymbol{B}_{+} \boldsymbol{u}^{i}(t) \leq \boldsymbol{1}_{|V|}$$
(12)

を満たす. (12) 式に i=I, (5), (7) 式を代入すると (6) 式が導けるため, (11) 式は (6) 式を満たす.

#### 3.5 探索行動記録

移動完了後、エージェントは割り当てられた探索ユニットに移動経路を記録する。時刻 t におけるユニットi の探索記録を $\hat{u}^i(t) \in \{0,1\}^{|E|}$  と表す。 $\hat{u}^i(t)$  の  $\gamma(x,y)$  番目の要素を $\hat{u}^i_{x,y}(t)$  と表す。時刻 t におけるユニットi の探索記録が $e_{xy} \in E$  の通過を表す場合。 $\hat{u}^i_{x,y}(t)$  は 1 となる。そうでない場合, $\hat{u}^i_{x,y}(t)$  は 0 となる。時刻 t においてユニットi が割り当てられるエージェントの ID を $\hat{\mu}^i(t) \in \{1,2,...,I\}$  と表す。エージェントi は $\hat{u}^{\mu^i(t)}(t-1)$ を

$$\hat{\boldsymbol{u}}^{\mu^{i}(t)}(t-1) = \boldsymbol{u}^{i}(t-1) + \tilde{\boldsymbol{u}}^{i}(t-1)$$
(13)

と記録する. 制約  $\mathbf{c}$ ) を考慮して本論文では  $\hat{m{u}}^i(t)$  に  $\hat{m{u}}^i(t)^{\mathrm{T}}\mathbf{1}_{|E|} \leq 1$  の制約を設ける. また,

$$\boldsymbol{z}^{\dot{\mu}^{\dot{i}}(t+1)}(t+1) = \boldsymbol{z}^{\dot{\mu}^{\dot{i}}(t)}(t) + (\boldsymbol{B}_{+} - \boldsymbol{B}_{-})\dot{\boldsymbol{u}}^{\dot{i}}(t) \quad (14)$$

を満たす $\hat{u}^i(t)$ を連続的な移動記録として扱い、

$$\grave{\boldsymbol{u}}^{\grave{i}}(t) \leq \hat{\boldsymbol{u}}^{\grave{\mu}^{\grave{i}}(t)}(t) \tag{15}$$

を満たす $\hat{u}^{i}(t)$ を経路計画に沿った移動記録として扱う.

# 4. 情報共有

本章ではエージェント間での情報の送受信と地図情報 共有機能について述べる.

エージェントの通信可能範囲を $d\in\mathbb{N}$ と表す。エージェントの間の距離はMにおける,各エージェントが存在するノード間の距離で表す.有向グラフにおける二つのノード間の距離は二つのノードを結ぶ最短経路の辺数である.エージェントは各時刻において距離がd以下となるエージェントと通信する.時刻tにおけるエージェント間の通信状態を $C(t)\in\{0,1\}^I$ と表す.C(t) のi 行j 列要素を $c_j^i(t)$  と表す.時刻t においてエージェントi とエージェントj が互いの通信可能範囲内に存在し通信を行う場合, $c_j^i(t)$  は1 となる.以降ではエージェントj がエージェントi からある情報 $\omega^i(t)$  を受信した場合,受信情報を元情報と区別するために $\omega^j(t,i)$  と表記する.このとき, $\omega^j(t,i)$  は $\omega^i(t)$  に対して

$$\bar{\omega}^j(t,i) = c_i^i(t)\omega^i(t) \tag{16}$$

と与えられる.

エージェントは (3), (4) 式で求めた地図情報を共有し、取得した地図情報を重ね合わせることで  $\hat{w}^{\mu^i(t)}(t)$ と  $\hat{q}^{\mu^i(t)}$  を決定する。時刻 t においてエージェント i が取得した地図情報の内,ID が j 以下のエージェントから取得した情報の重ね合わせによって得られる迷路の形状情報と検知状況情報をそれぞれ  $\check{w}^i(t,j) \in \{0,1\}^{|E|}$ ,  $\check{q}^i(t,j) \in \{0,1\}^{|E|}$  と表す。 $\check{w}^i(t,j)$  と  $\check{q}^i(t,j)$  はそれぞれ

$$\check{\boldsymbol{w}}^{i}(t,j) = \left(\boldsymbol{I} - \bar{\boldsymbol{Q}}^{i}(t,j)\right) \check{\boldsymbol{w}}^{i}(t,j-1) + \bar{\tilde{\boldsymbol{w}}}^{i}(t,j) \quad (17)$$

$$\check{\mathbf{q}}^{i}(t,j) = \left(\mathbf{I} - \bar{\mathbf{Q}}^{i}(t,j)\right)\check{\mathbf{q}}^{i}(t,j-1) + \bar{\mathbf{q}}^{i}(t,j) \tag{18}$$

と求まる. ただし,

$$\check{\boldsymbol{w}}^{i}(t,0) = \tilde{\boldsymbol{w}}^{i}(t), \qquad \qquad \check{\boldsymbol{q}}^{i}(t,0) = \boldsymbol{q}^{i}(t) \tag{19}$$

とする.  $ar{Q}^i(t,j) \in \{0,1\}^{|E| \times |E|}$  は  $\gamma(x,y)$  番目の要素に  $ar{q}^i(t,j)$  の  $\gamma(x,y)$  番目の要素をもつ対角行列であり、アークに関する情報から  $ar{q}^i(t,j)$  上で状態 "既知"に分類される範囲の情報を抽出する.  $\hat{w}^{\mu^i(t)}(t)$  と  $\hat{q}^{\mu^i(t)}(t)$  は

$$\hat{\boldsymbol{w}}^{\mu^{i}(t)}(t) = \check{\boldsymbol{w}}^{i}(t,I), \qquad \hat{\boldsymbol{q}}^{\mu^{i}(t)}(t) = \check{\boldsymbol{q}}^{i}(t,I)$$
 (20)

と与えられる.

# 5. デッドロック検知・回復

本論文ではエージェントの移動による区画占有の解消を区画占有の自然解消とし、自然解消の見込みがない区画占有群をデッドロックとして扱う。提案するデッドロック検知手法は文献 [5] のデッドロック予測手法を迷路の分散協調探索用に拡張したものである。文献 [5] で

はエージェントの移動要求に対し経路計画結果と経路の 占有状況をもとに Wait-for graph を作成し、Wait-for graph の形状からデッドロックを予測している。本章で は3.4節で示したエージェント間での衝突回避の結果を 利用したデッドロック検知、原因特定および回復動作に ついて述べる。

### 5.1 デッドロック検知

(10)、(11) 式より、以降では $p_j^i(t)$ が1となることをエージェントjの区画占有によるエージェントiに対する進路妨害として扱う。デッドロック検知にはエージェント間での進路妨害の情報を利用する。エージェントは他のエージェントから進路妨害の情報を取得し、取得した情報をもとに有向グラフを用いて個別にデッドロックの検知に取り組む。時刻tにおいてエージェントiが作成する有向グラフを $G^i(t)=(N,A^i(t))$ と表す。 $G^i(t)$ はエージェントを表すノードの有限集合 $N:=\{n_1,n_2,...,n_I\}$ とエージェント間での進路妨害を表すアークの有限集合 $A^i(t)$ で構成される。 $n_j \in N$  はエージェントjと対応し、 $a_{jk}=(n_j,n_k)\in A^i(t)$  はエージェントkのエージェントjに対する進路妨害を表す。 $A^i(t)$  は

$$A^{i}(t) = \sum_{j=1}^{I} \sum_{k=1}^{I} f(t, i, j, k)$$

$$f(t, i, j, k) = \begin{cases} \{a_{jk}\}, & \text{if } \bar{p}_{k}^{i}(t, j) = 1\\ \emptyset, & \text{if } \bar{p}_{k}^{i}(t, j) = 0 \end{cases}$$
(21)

と求められる.  $G^i(t)$  において  $n_j$  の出次数 outdeg $(n_j)$  はエージェント j が受ける進路妨害の数を表す. また、入次数 indeg $(n_j)$  はエージェント j による進路妨害の数を表す. 本論文では出次数、入次数が 0 となるノードをそれぞれ出端点、入端点と表記する. 制約 c) および (9), (10), (21) 式より

$$\operatorname{outdeg}(n_j) \le 1, \qquad \forall n_j \in N$$
 (22)

が成立する.

(11), (21) 式より outdeg $(n_i) = 0$  はエージェント j が 進路妨害を受けておらず、移動可能であることを表す. したがって、 $G^{i}(t)$  における出端点に対応するエージェ ントによる区画占有は自然解消される見込みがあると いえる. つぎに,  $G^i(t)$  から出端点を削除し, 出端点に 対応するエージェントによる区画占有がすべて解消され た状況について考える. ノードの削除は該当するノード と該当するノードに接続するすべてのアークの削除に よって達成される. 出端点を削除したとき, グラフの形 状によっては新たな出端点が発生する. 新たに発生した 出端点は区画占有の解消により移動が可能になったエー ジェントに対応し、対応するエージェントによる区画占 有が自然解消される可能性があることを示す. そこで, Algorithm 1にしたがって新たな出端点の発生がなくな るまで出端点の削除を繰り返す. このとき得られるグラ フを $G_d^i(t) = (N_d^i(t), A_d^i(t))$ と表す.  $G^i(t)$ のノード数が I 個であることから、Algorithm 1 では多くても I 回の出端点の削除によってすべての出端点が削除される。このとき、(22) 式より  $G_a^i$  について

$$\operatorname{outdeg}(n_j) = 1, \qquad \forall n_j \in N_d^i(t)$$
 (23)

が成り立つ. (23) 式は  $G_d^i(t)$  上のノードに対応するエージェントによる区画占有が自然解消不可能であることを意味する. したがって,  $a_{jk} \in A_d^i(t)$  で表される区画占有がデッドロックに相当する. (23) 式より, デッドロックが検知された場合  $G_d^i(t)$  は閉路をもつ.

# Algorithm 1 Detect deadlock of agent $\Delta_i$

```
1: Input G^i(t) as G = (N_a, A_a)
 2: N'_q = N_q;
 3: while N'_q \neq \emptyset do
       N_q' = \emptyset
       for j = 1; j <= I; j ++ do
 5:
          if n_i \in N_q && outdeg(n_i) == 0 then
 6:
 7:
             N_q' = N_q' \cup n_j;
          end if
 8:
 9:
       end for
       N_a = N_a \setminus N_a';
10:
11: end while
12: return G_d^i(t) = G
```

### 5.2 デッドロック原因の特定

検知されたデッドロック  $G_a^i(t)$  の解消のため、デッド ロックの原因となるエージェントを特定する.  $G_d^i(t)$  に おいて  $indeg(n_i)$  はエージェント i による進路妨害を原 因にもつ、自然解消の見込みがない区画占有の数を表す. したがって、入端点に対応するエージェントによる進路 妨害はデッドロックの原因とはならない. つぎに. グラ フ上から入端点を削除し,残った区画占有の原因につい て考える. このとき, グラフの形状によっては新たな入 端点が発生する. 新たな入端点に対応するエージェント による進路妨害は、残った区画占有の原因とはならない. そこで、Algorithm 2に従って、新たな入端点の発生が なくなるまで入端点の削除を繰り返す. このとき得られ るグラフを $G_c^i(t) = (N_c^i(t), A_c^i(t))$ と表す. Algorithm 1 と同じく、Algorithm 2も多くても I 回の入端点の削除 によってすべての入端点が削除される. 入端点の定義よ り、入端点の削除において削除されるアークは削除され るノードを始点にもつアークに限られる. したがって, 削除後に残ったノードに関して出次数の変化は発生しな い. このとき,  $G_c^i(t)$  について (23) 式より

 $\operatorname{indeg}(n_i) \ge 1$ ,  $\operatorname{outdeg}(n_i) = 1$ ,  $\forall n_i \in N_c^i(t)$  (24)

が成り立つ. (24) 式は  $G_c^i(t)$  上のノードに対応するエージェントによる進路妨害が自然解消不可能な区画占有の原因となっていることを表す. また, (24) 式より

$$indeg(n_j) = outdeg(n_j) = 1, \quad \forall n_j \in N_c^i(t) \quad (25)$$

が成り立つため、 $G_c^i(t)$  はオイラーグラフとなる。また、 $G_c^i(t)$  の各連結成分はオイラー閉路となる。

## Algorithm 2 Identify agents causing deadlocks

```
1: Input G_d^i(t) as G = (N_q, A_q)
 2: N'_q = N_q;
 3: while N_q \neq \emptyset do
       N_g' = \emptyset
       for j = 1; j <= I; j ++ do
          if n_j \in N_q && indeg(n_j) == 0 then
 6:
             N_q' = N_q' \cup n_j;
 7:
 8:
          end if
       end for
 9:
       N_g = N_g \setminus N_g';
10:
11: end while
12: return G_c^i(t) = G
```

## **5.3** デッドロック回復動作

デッドロックからの回復はその原因となるエージェントの経路計画結果の変更によって達成される。本論文では、デッドロックの原因となるエージェント間での役割交換と探索の引継ぎによるエージェントの経路計画結果を変更する手法を提案する。役割交換と探索の引継ぎはエージェント間での探索ユニットの交換と、疑似移動による移動後の状態の引継ぎによって構成される。探索ユニットの交換によりエージェントがもつ役割が変化するため、経路計画結果を変化させることができる。さらに、連続的かつ経路計画結果に沿った移動記録が得られるように探索ユニットの交換と疑似移動を行うことで、デッドロック解決のためのエージェントの移動経路変更を抑える。(5),(11),(21),(25)式および $G_c^i(t)\subseteq G^i(t)$ より

$$o^{j}(t) = z^{j}(t), \qquad \forall n_{i} \in N_{c}^{i}(t)$$
 (26)

が成り立つ. (1), (7), (10), (21), (26) 式より

$$\boldsymbol{z}^{k}(t+1) = \boldsymbol{z}^{j}(t) + (\boldsymbol{B}_{+} - \boldsymbol{B}_{-})\hat{\boldsymbol{u}}^{j}(t), \qquad (27)$$
$$\forall a_{jk} \in A_{c}^{i}(t)$$

が成り立つ. そこで、デッドロック  $G_d^i(t)$  からの回復における探索ユニットの交換を

$$\mu^k(t+1) = \mu^j(t), \qquad \forall a_{jk} \in A_c^i(t)$$
 (28)

疑似移動動作を

$$\tilde{\boldsymbol{u}}^k(t) = \hat{\boldsymbol{u}}^j(t), \qquad \forall a_{ik} \in N_c^i(t)$$
 (29)

と与える. (13)~(15), (27) 式より (28), (29) 式で表す探索の引継ぎでは連続的かつ経路計画結果に沿った移動経路が得られる. 探索ユニットの交換は他のエージェントの探索ユニットの引き受けと. 他のエージェントへの探

索ユニットの引き渡しで構成される。 (28) 式はエージェント j による探索ユニットの引き渡しとエージェント k による探索ユニットの引き受けと表せる。探索ユニットの交換において, $G_c^i(t)$  上のノードの出次数と入次数は それぞれ対応するエージェントにおける探索ユニットの引き渡しと引き受けの数を表す。 (25) 式より (28) 式に沿った探索ユニットの交換では,各エージェントの探索 ユニットの引き渡し数と引き受け数が一致するため探索 ユニットの重複割当てや消失は発生しない。

(28), (29) 式で定義されるデッドロックからの回復動作を実現するためには、 $G_c^i(t)$  に含まれるノードに対応するすべてのエージェントが $G_c^i(t)$  の情報を保有する必要がある。そこで、エージェントi は $G_c^i(t)$  の情報から探索引継ぎ指令 $\mathbf{R}^i(t) \in \{0,1\}^{I \times I}$  を作成する。 $\mathbf{R}^i(t)$  の j 列要素を $\mathbf{r}^i_{j,k}(t)$  と表し、 $\mathbf{r}^i_{j,k}(t)$  は表す。 $\mathbf{r}^i_{j,k}(t)$  は

$$r_{j,k}^{i}(t) = \begin{cases} 1, & \text{if } a_{kj} \in A_c^i(t) \\ 0, & \text{otherwise} \end{cases}$$
 (30)

と与えられる。エージェントjは受信した情報をもとに探索引継ぎ対象 $\sigma^{j}(t)$ を

$$\sigma^{j}(t) = \sum_{i=1}^{I} \check{\sigma}^{j}(t,i)$$

$$\check{\sigma}^{j}(t,i) = \check{e}^{j}(t,i)[1 \ 2 \ \dots \ I]\bar{r}^{j}_{j}(t,i)$$

$$\check{e}^{j}(t,1) = 1$$

$$\check{e}^{j}(t,i+1) = \begin{cases} 1, & \text{if } \check{e}^{j}(t,i) = 1 \land \check{\sigma}^{j}(t,i) = 0 \\ 0, & \text{otherwise} \end{cases}$$
(31)

と決定する.  $\check{e}^j(t,i)$  は探索引継ぎ命令を重複して受け取ることを防止するための変数である. このとき,

$$\tilde{\boldsymbol{u}}^{j}(t) = \begin{cases} \hat{\boldsymbol{u}}^{\sigma^{j}(t)}, & \text{if } \sigma^{j}(t) \neq 0\\ \mathbf{0}, & \text{if } \sigma^{j}(t) = 0 \end{cases}$$

$$\mu^{j}(t+1) = \begin{cases} \mu^{\sigma^{j}(t)}, & \text{if } \sigma^{j}(t) \neq 0\\ \mu^{j}(t), & \text{if } \sigma^{j}(t) = 0 \end{cases}$$
(32)

とすることで (28), (29) 式が達成される.

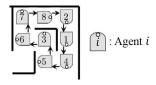
 $\hat{u}^i(t) = \mathbf{0}$  が満たされる場合、エージェントi による区画占有は自然解消が見込めない。このとき、エージェントi は自身が進路妨害しているエージェントの中から ID の小さいものを優先して探索の引継ぎを行う。

#### 5.4 性能評価

本論文では、 $G_c^i(t)$  における各連結成分のノード数をデッドロックの規模として扱う。規模 8 のデッドロックの例と d=2 としたときの  $G^2(t)$  と  $G^3(t)$  を Fig. 2 に示す。Fig. 2 ではエージェント 3 のみがデッドロックを検知したため (b) は閉路をもたず,(c) のみが閉路をもつ。このことから,d=2 とした場合,規模 8 のデッドロックの検知は保証されないことがわかる。

ここで、規模  $d_{\alpha}$  のデッドロックに対し、M におけるデッドロックの原因となるエージェント間の最大距離  $d_{\beta}$ 

を考える。4章で定義した有向グラフ上での距離を用いると、g個のノードで構成されるオイラー閉路においてノード間の最大距離  $d_g$  は  $2d_g \leq g$  を満たす。また、(27) 式および制約 c) より、 $G_c^i(t)$  上で隣接する二つのノードに対応するエージェント間の M における距離は 1 となる、そのため、 $2d_\beta \leq d_\alpha$  である。 $d \geq d_\beta$  となるときデットロックが検知されることから、規模  $d_\alpha$  のデッドロック検知の十分条件は  $2d \geq d_\alpha$  と表される。



(a) Position of agents and path plans

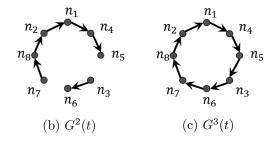


Fig. 2 Deadlock of scale 8

# 6. 数值実験

### 6.1 デッドロック検知・回復手法の動作検証

提案手法によるデッドロック検知・回復手法の動作検証として数値実験を行う. Fig. 3 に数値実験に利用した迷路とエージェントの初期状態を示す.

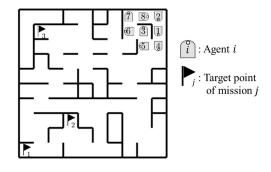


Fig. 3 Initial layout of agents

本実験ではエージェントの向きをエージェントの状態として扱う。エージェントの向きはエージェント記号先端の丸みの方向で表す。探索システムはd=2である8台のエージェントで構成され,三つの目的地到達ミッションと周辺環境把握ミッションをもつ。ユニット $1\sim 3$ はそれぞれのIDに対応する目的地到達ミッションをもち,足立法[11]に従って割り当てられたエージェントの移動経路を決定する。ユニット $4\sim 8$ は周辺環境把握ミッションをもち,トレモー法[12]に従って割り当てられたエージェントの移動経路を決定する。実験結果に再現性をもたせるためトレモー法では分岐点における経路

選択にエージェントの向きに応じた優先順位を設けてい る. エージェントの初期状態と経路計画結果およびデッ ドロック検出結果はFig.2と一致し、t=1において規模 8のデッドロックが発生する.

Fig. 4 に探索の結果を, Fig. 5 に時刻 20 までの探索ユ ニットの割り当て状況を示す. Fig.4 において灰色の線 は迷路の実形状を表し、黒色の線は $\hat{w}^{8}(t)$ を表す。

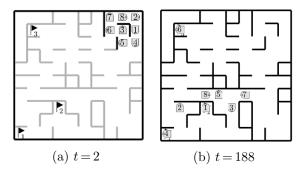


Fig. 4 Exploration results

Fig. 3と Fig. 4(a) ではエージェント2の向きが変わっ ていることがわかる. これは、デッドロック回復のために 行った疑似移動動作によるものである. また, Fig. 4(b) より全目的地到達ミッションと周辺環境把握ミッション の達成が確認できる.

	agent 1	1	44	4-4-	4-4	4-4	4-4-	4-4-	4-4-	4-4-	4-4-	4	
agent ID	agent 2	2)	1	1)-(1)-(	1)-(1)-	1-1-	1-1-	1)-(1)-	1-1-	1)-(1)-	1)-(1)-	1	
	agent 3	3/	6-6	6-6-	6-6-	6-6-	6-6-	6-6-	6-6-	6-6-	6-6-	6	
	agent 4	4	5+5+	5-5-	5-5-	<b>(5)-(5)</b> -	5-5-	5-5-	5-5-	5-5-	5-5-	5	
	agent 5	(5)\(	3+3+	3-3-	3-3-	3-3-	3-3-	3-3-	3-3-	3-3-	3 3	8	
	agent 6	6	$\mathfrak{D}_{\mathcal{T}}$	<u>7</u> -7-	<u>I</u> -I	7-7-	7-7-	<u> </u>	7	(T)-(T)-	⑦ <del>;</del> ⑦⟨	$\bigcirc$	
	agent 7	T)	8 <del>;</del> 8	8-8	8-8	8-8-	8-8-	8-8	8-8	82	2 <del> </del> 2}^	2	
	agent 8	8	2 2	2-2-	2-2-	2-2-	2-2-	2-2-	22	28	8 8	3	
		0	2	4	6	8	10	12	14	16	18	20	
	time t												

Fig. 5 Unit

Fig. 5はエージェントに割り当てられた探索ユニット の ID を丸の内側の数字で、探索ユニットの交換が行わ れた時刻を点線で表す. t=1 における探索ユニットの交 換を Fig. 2 と比較すると、探索ユニットの交換が Fig. 2(c) のアークの方向に従って行われていることが確認で きる. また, Fig. 2(a) より Fig. 4(a) における各エー ジェントの向きは、t=1における各エージェントの探索 引継ぎ対象の経路計画結果に従っていることがわかる. これらのことから、提案アルゴリズムによるデッドロッ ク回復において探索ユニットの再割り当てと探索引継ぎ は、それぞれ(28)式と(29)式を満たしていることがわ かった. 規模8のデッドロックはFig.2と同様にエージェ ント3によって検出され、探索指令に基づく探索の引継 ぎによって解消された. この結果より. 提案アルゴリズ ムを用いて分散協調探索におけるデッドロックの検知・ 回復が可能となることが示せた.

#### 6.2 提案手法の特徴

つぎに、提案手法によるデッドロック検知・回復が探 索に与える影響を考察する. Fig. 6 に本実験の概要図を 示す. 探索対象は10×10の2次元迷路であり. 有向グ ラフ表現Mでは|V|=100, |E|=360となる.

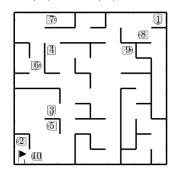


Fig. 6 Simulation outline

ユニット1は迷路右上から左下の目的地までの移動の ミッションをもち、足立法に従って割り当てられたエー ジェントの移動経路を決定する. その他の探索ユニット はミッションをもたず、あらかじめ与えられた経路に沿っ た移動を役割としてエージェントに与える。エージェン ト数を1,10,20,30,...,80と変化させ、それぞれ単体での 目的地探索との比較によりエージェント数が探索効率に 与える影響を評価する. 本実験では常にすべてのエー ジェントが情報共有可能な状況を想定し. エージェント の通信可能範囲をd=18としている。また、探索対象の 形状やミッションをもたないエージェントの移動経路が 探索結果に与える影響を考慮し、探索対象とエージェン トの移動経路を変更し10回の実験を行っている。t=200までに目的地到達が達成されなかった場合は探索失敗と して扱う.

評価項目は探索効率と稼働率である. 探索効率は探索 所要時間Tと移動経路長Lで構成される。Tはユニット 1の目的地到達までに経過した時間. L はユニット1に 記録された移動経路の長さを表す. Lは

$$L = \sum_{t=1}^{T} \dot{\boldsymbol{u}}^{1}(t)^{\mathrm{T}} \mathbf{1}_{|E|}$$
(33)

と与えられる. 稼働率はエージェント稼働率 $\alpha$ とユニッ ト稼働率 $\alpha$ で構成される。 $\alpha$  は各時刻において区画移動 を行うエージェントの割合の平均値を表す. α は各時刻 において区画移動あるいは疑似移動動作を行うエージェ ントの割合の平均値を表す.  $\alpha$  と  $\hat{\alpha}$  はそれぞれ

$$\alpha = \frac{1}{TI} \sum_{t=1}^{T} \sum_{i=1}^{I} \mathbf{u}^{i}(t)^{\mathrm{T}} \mathbf{1}_{|E|}$$

$$\dot{\alpha} = \frac{1}{TI} \sum_{t=1}^{T} \sum_{i=1}^{I} \left( \mathbf{u}^{i}(t) + \tilde{\mathbf{u}}^{i}(t) \right)^{\mathrm{T}} \mathbf{1}_{|E|}$$
(34)

$$\dot{\alpha} = \frac{1}{TI} \sum_{t=1}^{T} \sum_{i=1}^{I} \left( \boldsymbol{u}^{i}(t) + \tilde{\boldsymbol{u}}^{i}(t) \right)^{\mathrm{T}} \mathbf{1}_{|E|}$$
 (35)

と与えられる. ここで、エージェントの位置の偏りを考 慮しない場合のエージェント稼働率の理想値を考える. エージェント稼働率の理想値は  $2I \le |V|$  となる場合と 2I > |V| となる場合で変化する.  $2I \le |V|$  となる場合,  $\sum_{i=1}^{I} \boldsymbol{u}^{i}(t)^{\mathrm{T}} \mathbf{1}_{|E|}$  は制約 c) より

$$\sum_{i=1}^{I} \boldsymbol{u}^{i}(t)^{\mathrm{T}} \mathbf{1}_{|E|} \leq I \tag{36}$$

を満たす. つぎに 2I > |V| となる場合について考える. M においてすべてのアークには終点が一つずつ設定されるため.

$$\boldsymbol{u}^{i}(t)\mathbf{1}_{|E|} = \left(\boldsymbol{B}_{+}\boldsymbol{u}^{i}(t)\right)^{\mathrm{T}}\mathbf{1}_{|V|} \tag{37}$$

となる. (2), (5), (37) 式より  $u^i(t)\mathbf{1}_{|E|}$  は $o^i(t)$  に対して

$$\boldsymbol{o}^{i}(t)^{\mathrm{T}} \mathbf{1}_{|V|} = 1 + \boldsymbol{u}^{i}(t)^{\mathrm{T}} \mathbf{1}_{|E|}$$
(38)

を満たす。このとき、(6)式の両辺を転置し、右から  $\mathbf{1}_{|V|}$  をかけると

$$\sum_{i=1}^{I} \boldsymbol{o}^{i}(t)^{\mathrm{T}} \mathbf{1}_{|V|} \leq \mathbf{1}_{|V|}^{\mathrm{T}} \mathbf{1}_{|V|} = |V|$$
(39)

が得られる。(39)式に(38)式を代入すると

$$\sum_{i=1}^{I} 1 + \left(\sum_{i=1}^{I} \boldsymbol{u}^{i}(t)^{\mathrm{T}} \mathbf{1}_{|E|}\right) \leq |V|$$

$$\tag{40}$$

となる. (36), (40) 式より (34) 式に示すエージェント稼働率の理想値  $\hat{\alpha}$  は M とエージェント数 I に対して

$$\hat{\alpha} = \begin{cases} 1, & \text{if } 2I \leq |V| \\ \frac{|V| - I}{I}, & \text{if } 2I > |V| \end{cases}$$

$$\tag{41}$$

と求まる.

探索効率と稼働率のエージェント数増加に対する推移をそれぞれ Fig. 7. Fig. 8 に示す.

Fig. 7では単体での目的地探索時の探索効率の値を1とした場合の、各エージェント数における探索効率の値を表示している。エージェント数が増加すると移動経路長は減少する傾向にある。このことから、各エージェントが全体の達成目標を考慮して経路計画を行わない状況でも、情報共有によって各エージェントの探索効率を改善することができることがわかる。また、衝突やデッドロックの発生リスクが高まるエージェント数が多い状況においても、移動経路長は増加していない。ただし、エージェント数が20以上の場合には、エージェント数の増加に伴い探索所要時間が増加している。

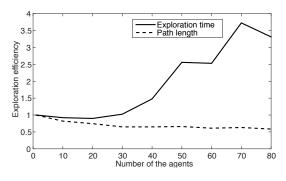


Fig. 7 Exploration efficiency

Fig. 8 からエージェント稼働率とユニット稼働率はどちらもエージェント稼働率の理想値と比べて低くなることがわかる. Fig. 7 で確認されたエージェント数増加に伴う探索所要時間の増加は、稼働率の低下によるものであると思われる. これらの結果から提案アルゴリズムを用いた迷路探索では、稼働率の確保と比べて移動経路長の抑制が優先されるといえる. これは、デッドロック解決によるエージェントの移動経路の変更を抑えるという目標に沿った結果といえる.

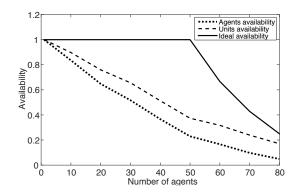


Fig. 8 Exploration efficiency

# 6.3 エージェント数と通信可能範囲の影響

つぎに、エージェント数や通信可能範囲が協調探索の成否に与える影響を考察する。通信可能範囲をd=1,2,3,18と変化させ、6.2節と同一の迷路、エージェント台数、移動経路を設定した際の探索の成否を確認する。6.2節と同様に、t=200までにユニット1の目的地到達が達成されなかった場合を探索失敗とする。各エージェント数、通信可能範囲における探索失敗確率をFig.9に示す。

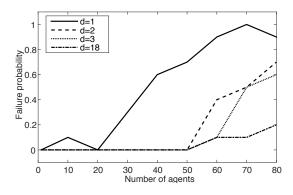


Fig. 9 Exploration failure probability

Fig. 9では通信可能範囲が狭くなるにつれ,各エージェント数における探索失敗確率の増加が確認された.探索失敗確率の増加の要因は各通信可能範囲における解決不可能な規模のデッドロックの発生である.通信可能範囲をd=18とした場合は全エージェントの通信可能範囲が迷路全域にわたるため,発生したすべてのデッドロックの検出・回復が可能である.本実験では通信可能範囲を

d=1 とした場合、40 台以上のエージェントによる協調探索を成功させることが困難になるという結果になった。

#### 7. おわりに

本論文では通信可能範囲に制限をもつエージェント群による迷路の協調探索手法における情報共有の手段とデッドロック解決手法を提案した. 対象となる探索システムは複数のエージェントとエージェントに役割を与える探索ユニットで構成される. エージェントがもつ迷路探索機能は単体で扱う基本機能と地図情報共有とデッドロック解決で構成される協調機能に分割される. 基本機能に役割に応じた経路計画手法の使い分けと疑似移動を導入し, 基本機能の規格化と協調機能の作成および数理モデル化に取り組んだ. 提案手法はエージェント間での情報共有によって通信可能範囲の制限下におけるデッドロック検知と, 検知されたデッドロックの解消を実現する. 作成した数理モデルをもとに通信可能範囲と検知が保証されるデッドロックの規模の関係を示した.

また、数値実験によって提案手法によるデッドロック解決機能の動作検証を行った.併せて、提案する探索アルゴリズムがもつ特徴を迷路探索の効率とエージェントの稼働率の面から評価し、通信可能範囲とエージェント数が探索の成否に与える影響について考察した.

今後の課題としてエージェントの台数や通信可能範囲が探索効率に与える影響の定量的評価が挙げられる。同時にデッドロック解決手法の簡略化として、通信する情報量や通信回数の削減に取り組む。本研究は JSPS 科研費 17H06293 の助成を受けたものである。

# 参考文献

- [1] K. Knight: Are many reactive agents better than a few deliberative ones?; Proceedings of the Thirteen International Joint Conference on Artificial Intelligence, pp. 432–437 (1993)
- [2] 横尾:淘汰を用いたマルチエージェント実時間探索の高速化:協調探索への競争の導入;コンピュータソフトウェア, Vol. 14, No. 4, pp. 47–55 (1997)
- [3] R. S. Bharathi, R. Priyadharshni and S. Ramesh: Artificial bee colony algorithm to find optimum path for mobile agents in wireless sensor networks; *IEEE International Conference on Advanced Communication Control and Computing Technologies*, pp. 842–846 (2014)
- [4] E. H. Kivelevich and K. Cohen: Multi-agent maze exploration; Journal of Aerospace Computing, Information, and Communication, Vol. 7, pp. 391–405 (2010)
- [5] L. Kalinovcic, T. Petrovic, S. Bogdan and V. Bobanac: Modified Banker's algorithm for scheduling in multi-AGV systems; *IEEE International Conference on Automation Science and Engineering*, pp. 351–356 (2011)
- $[6]\,$  N. Wu and M. C. Zhou: AGV routing for conflict

- resolution in AGV systems; International Conference on Robotics & Automation, pp. 14–19 (2003)
- [7] S. A. Reveliotis: Conflict resolution in AGV systems; *IIE Transactions*, Vol. 32, pp. 647–659 (2000)
- [8] 小泉, 増山:梯子型走行経路を用いた AGV システムに おけるデッドロック回復問題の計算複雑さ;システム制 御情報学会論文誌, Vol. 23, No. 5, pp. 91–104 (2010)
- [9] 望月, 澤田, 新: 異なる役割を持つエージェント群による迷路探索の研究; 計測自動制御学会 システム・情報部門 学術講演会 予稿集 (2016)
- [10] A. Siberschatz, P. B. Galvin and G. Gagne: Operating System Concepts, Wiley Publishing (2005)
- [11] B. H. Kazerouni, M. B. Moradi and P. H. Kazerouni: Variable priorities in maze-solving algorithms for robot's movement; *IEEE International Conference on Industrial Informations*, Vol. 6, pp. 181–185 (2003)
- [12] O. Ore: *Theory of Graphs*, American Mathmatical Society (1962)

#### 著者略歴

# が 優加理



1994年3月10日生. 2016年3月電気通信大学情報理工学部知能機械工学科卒業. 同年4月電気通信大学大学院情報理工学研究科機械知能システム学専攻博士前期課程に入学し現在に至る.

# 澤田 賢治(正会員)



2009年大阪大学大学院工学研究科機械工学専攻博士後期課程修了.同年電気通信大学システム工学科助教,2010年同大学知能機械工学科助教,2015年同大学i-パワードエネルギー・システム研究センター准教授となり現在に至る.ハイブリッドシ

ステムや制御系セキュリティに関する研究に従事. 博士(工学). 計測自動制御学会, 電子情報通信学会, 電気学会, IEEE会員. 2016年より制御システムセキュリティセンター顧問.

# 新 誠 一 (正会員)



1980年東京大学大学院工学系研究科修士課程修了.同年,同大学工学部計数工学科助手.1987年工学博士(東京大学).同大学講師を経て,1988年筑波大学電子·情報工学系助教授,1992年東京大学工学部助教授,2001年同大学情報理工学系研究

科助教授,2006年電気通信大学教授.同年,計測制御エンジニア.1991年,93年,98年計測自動制御学会論文賞,92年同賞武田賞,2006年同技術賞受賞.制御理論を中心に広く工学全体に興味を持つ.日本応用数理学会評議員,(財)製造科学技術センター評議員,製造業XML推進協議会運営委員長,FAオープン推進協議会電子タグ活用専門委員会主査,2012年より制御システムセキュリティセンター理事長.