分散協調制御システムに基づく 迷路探索問題における通信手段と探索効率の関係*

望月優加理[†]・澤田 賢治[‡]・新 誠一[†]

A Relationship between the Communication Protocol and Exploration Efficiency on Maze Exploration via Distributed Collaborative Control System*

Yukari MOCHIZUKI[†], Kenji SAWADA[‡] and Seiichi SHIN[†]

In cooperative maze exploration via multi-agent systems, deadlock resolution methods influence exploration efficiency. This paper evaluates the relation between communication protocols and exploration efficiency. This paper proposes three communication protocols for deadlock resolution on maze exploration via distributed collaborative control system considering communicable range of agents. The agents share different information depending on the communication protocols for deadlock resolution. The transmission route of the information is affected by the communication protocols for deadlock resolution. The first result is to show a relationship between communicable range and performance of deadlock detection and communication load as the characteristics of each proposed communication protocols. The second result is to derive the relation between flow quantity and density for some network structure. Using the two results we evaluate influence of communication protocols on exploration efficiency.

1. はじめに

マルチエージェントシステムでは自律的に意思決定を行う複数のエージェントが協力することで、より複雑な機能やタスク処理効率の向上を実現する。文献[1]ではマルチエージェントシステムに期待される特徴として、システム全体の可用性の確保[2]とエージェント間での協力やタスクの並列処理によるタスク処理の効率化[3–5]を挙げている。著者らは文献[6]において通信可能範囲内での情報共有にもとづく、異なる役割を持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・回復手法を提案している。迷路の協調探索におけるデッドロックの解決は、システム全体の可用性の確保に相当する。また、デッドロックの解決手法は各エージェント

のタスク消化までの移動経路長やエージェントの稼働率の面で、探索効率に影響を及ぼす。文献[7]によると移動順序の変更のみを用いたデッドロック回避[8]では、再経路計画によるデッドロック回避[9,10]や退避行動を含むデッドロック検知・回復[11]と比較して移動経路長の増加抑止が優先される。文献[6]のデッドロック検知手法はWait-for graphを用いたデッドロック発生予測手法[8]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。デッドロックからの回復動作はエージェント間での役割交換と探索引継ぎで構成される。探索引継ぎの導入により、役割の交換によるエージェントの移動経路の変更を抑制している。文献[6]のデッドロック検知手法において、デッドロック検知能力は通信可能範囲に依存する。そのため、エージェント数に対して通信可能範囲が十分に確保されない場合、発生したデッドロックを検知できず稼働率が大幅に低下し探索の継続が困難になる状況が見られている。

そこで、本論文では文献[6]のデッドロック解決手法に対する通信手段を3種提案し、通信可能範囲が不十分な状況における通信手段が迷路の探索効率に与える影響について考察する。はじめに、エージェントの迷路探索機能のうちデッドロック解決に関する機能の数理モデルを示す。つぎに、デッドロック解決アルゴリズムと、デッ

* 原稿受付 2018年3月30日

[†] 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-Communications; 1-5-1, Chofugaoka, Chofu city, Tokyo 182-8585, JAPAN

[‡] 電気通信大学 i-Powered Energy System Research Center, The University of Electro-Communications; 1-5-1, Chofugaoka, Chofu city, Tokyo 182-8585, JAPAN

Key Words: multi-agent system, communication protocol, flow quantity.

提案する通信手段についてデッドロックの正常な検知が保証される条件と、通信負荷を導出する。また、使用する通信手段が迷路探索の効率に与える影響を数値実験にもとづき考察する。以降では探索システムにおける i 番目 ($i=1,2,\dots,I$) 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 ($i=1,2,\dots,I$) 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. 迷路探索機能

本章ではエージェントがもつ迷路探索機能のうち、デッドロックの発生と解決に関する機能の数値モデルを示す。エージェントがもつ迷路探索機能は、それぞれのエージェントが単体で扱う基本機能と他のエージェントとの通信を必要とする情報共有機能に分けられる。変数の詳細な定義とデッドロックの解決に直接関係しない機能の数値モデルについては文献 [6] を参照されたい。

3.1 基本機能

本節では基本機能として、エージェントの自己位置同定、経路計画、移動、探索行動記録について述べる。

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を同定する。時刻 t におけるエージェント i の位置を $\mathbf{z}^i(t) \in \{0,1\}^{|V|}$ と表す。時刻 t におけるエージェント i の移動経路が $\mathbf{u}^i(t) \in \{0,1\}^{|E|}$ と与えられるとき、 $\mathbf{z}^i(t+1)$ は

$$\mathbf{z}^i(t+1) = \mathbf{z}^i(t) + (\mathbf{B}_+ - \mathbf{B}_-) \mathbf{u}^i(t) \quad (1)$$

と求まる。

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 $\hat{\mathbf{u}}^i(t) \in \{0,1\}^{|E|}$ と表す。

制約 e) より、 $\hat{\mathbf{u}}^i(t)$ は

$$\hat{\mathbf{u}}^i(t)^T \mathbf{1}_{|E|} \leq 1 \quad \wedge \quad \hat{\mathbf{u}}^i(t) \leq \mathbf{B}_-^T \mathbf{z}^i(t) \quad (2)$$

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。

経路計画決定後、エージェントは移動を開始する。制約 a) により、エージェントの移動のタイミングは同期される。制約 a) における移動は衝突回避のための停止行動や疑似移動動作を含む。疑似移動動作はエージェント間での探索引継ぎのための、特定の区画移動の結果を模した行動である。ここで、時刻 t におけるエージェント i の移動経路を (1) 式で用いた $\mathbf{u}^i(t)$ 、疑似移動動作を $\tilde{\mathbf{u}}^i(t) \in \{0,1\}^{|E|}$ と表す。移動においてエージェントは制約 b) に従って衝突回避を行う。衝突回避では他のエージェントの ID の取得が必要となる。ただし、衝突回避における情報共有は制約 c) に基づいて検知される同一

の区画に移動を試みるエージェント間でのみ行われる。そのため、ID を取得する段階において、情報共有を行うエージェント間の距離は通信可能範囲に対して十分に近いと仮定する。本論文ではエージェント間の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置くと、エージェント間での衝突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 $\sigma^i(t) \in \{0,1\}^{|V|}$ と表す。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 $p_j^i(t) \in \{0,1\}$ と表す。 $p_j^i(t)$ を j 番目の要素にもつ列ベクトルを $\mathbf{p}^i(t)$ と表す。

$\sigma^i(t)$ は

$$\sigma^i(t) = \mathbf{z}^i(t) + \mathbf{B}_+ \mathbf{u}^i(t) \quad (3)$$

と与えられ、制約 b) は

$$\sum_{i=1}^I \sigma^i(t) \leq \mathbf{1}_{|V|} \quad (4)$$

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 $\tilde{\sigma}^i(t,j) \in \{0,1\}^{|V|}$ と表す。 $\tilde{\sigma}^i(t,j)$ は優先順位を考慮して

$$\tilde{\sigma}^i(t,j) = \begin{cases} \mathbf{z}^j(t) & \text{if } j \geq i \\ \sigma^j(t) & \text{if } j < i \end{cases} \quad (5)$$

と与えられる。(3), (5) 式より (4) 式が満たされるとき $\tilde{\sigma}^i(t,j)$ は

$$\sum_{j=1}^I \mathbf{z}^j(t) \leq \sum_{j=1}^I \tilde{\sigma}^i(t,j) \leq \mathbf{1}_{|V|} \quad (6)$$

を満たす。 $\hat{\mathbf{u}}^i(t)$ の終点が $\tilde{\sigma}^i(t,j)$ 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、 $p_j^i(t)$ は

$$p_j^i(t) = \tilde{\sigma}^i(t,j)^T \mathbf{B}_+ \hat{\mathbf{u}}^i(t) \quad (7)$$

と求められる。(5), (7) 式による $\mathbf{p}^i(t)$ の導出には、 $\hat{\mathbf{u}}^i(t)$ の終点に存在するエージェントの現在地、および $\hat{\mathbf{u}}^i(t)$ の終点に移動するエージェントの移動先と ID の情報が必要となる。この条件は制約 c) と通信による ID の取得によって達成される。(4) 式を満たす $\mathbf{u}^i(t)$ として

$$\mathbf{u}^i(t) = \begin{cases} \hat{\mathbf{u}}^i(t) & \text{if } \sum_{j=1}^I p_j^i(t) = 0 \\ \mathbf{0}_{|E|} & \text{if } \sum_{j=1}^I p_j^i(t) = 1 \end{cases} \quad (8)$$

を用いる。ここで、 $\mathbf{0}_n$ はすべての要素が 0 である n 個の要素で構成される列ベクトルである。(7), (8) 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 のエージェント i に対する進路妨害とする。また、(2), (6), (7)

式より $p^i(t)$ は

$$p^i(t)^T \mathbf{1}_I \leq 1 \quad (9)$$

を満たす。

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 $\mu^i(t) \in \{1, 2, \dots, I\}$ と表す。

3.2 情報共有機能

情報共有のための機能として通信可能範囲を考慮したエージェント間の通信の数理モデルを示す。エージェント i は $z^i(t)$ が示す現在地を中心とする、通信可能範囲内に存在する他のエージェントとのみ情報共有を行う。

エージェントの通信可能範囲を $d \in \mathbb{N}$ と表す。また、時刻 t におけるエージェント j とエージェント k の距離を $\delta_m(t, j, k)$ 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。以降ではある有向グラフ $G = (N, A)$ におけるノード $n_i \in N$ と $n_j \in N$ の距離を $\delta_g(G, n_i, n_j)$ と表す。

時刻 t におけるエージェント i とエージェント j の通信状態を $c_j^i(t) \in \{0, 1\}$ 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 が通信による情報共有を行う場合、 $c_j^i(t)$ は 1 となる。そうでない場合、 $c_j^i(t)$ は 0 となる。また、時刻 t におけるエージェント間の通信状態を有向グラフ $\dot{G}_n(t) = (\dot{N}, \dot{A}_n(t))$ で表す。 $\dot{G}_n(t)$ はエージェントに対応するノードの有限集合 $\dot{N} := \{n_1, n_2, \dots, n_I\}$ とエージェント間の通信状態を表すアークの有限集合 $\dot{A}_n(t)$ で構成される。 $n_i \in \dot{N}$ はエージェント i と対応し、 $a_{ij} = (n_i, n_j) \in \dot{A}_n(t)$ は $c_j^i(t) = 1$ を表す。 $c_j^i(t)$ は

$$c_j^i(t) = \begin{cases} 1 & \text{if } \delta_m(t, i, j) \leq d \\ 0 & \text{otherwise} \end{cases} \quad (10)$$

と与えられる。(10)式より $\delta_m(t, i, j)$ と $\delta_g(\dot{G}_n(t), n_i, n_j)$ の間には

$$\delta_m(t, i, j) \leq d \delta_g(\dot{G}_n(t), n_i, n_j) \quad (11)$$

が成り立つ。以降では、エージェント i がエージェント j からある情報 $\omega^j(t)$ を受信した場合、受信情報を元情報と区別するために $\bar{\omega}^i(t, j)$ と表記する。このとき、 $\bar{\omega}^i(t, j)$ は $\omega^j(t)$ に対して

$$\bar{\omega}^i(t, j) = c_j^i(t) \omega^j(t) \quad (12)$$

と与えられる。ここで、(12)式は

$$\bar{\omega}^i(t, j) = \begin{cases} \omega^j(t) & \text{if } a_{ij} \in \dot{A}_n(t) \\ 0 & \text{otherwise} \end{cases} \quad (13)$$

を意味する。(12), (13)式よりエージェント i はエージェント j から情報の取得ができない場合、受信情報を 0 または零ベクトルとして扱う。

4. デッドロック解決アルゴリズム

4.1 デッドロック解決の手順

文献 [6] と同様にエージェント間の進路妨害関係にもとづいてデッドロックを解決する。以降では、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する。デッドロック解決アルゴリズムはデッドロックの検知・原因特定とデッドロックからの回復動作で構成される。

時刻 t におけるエージェント間の進路妨害を有向グラフ $\dot{G}_w(t) = (\dot{N}, \dot{A}_w(t))$ で表す。 $\dot{G}_n(t)$ と同様に $n_i \in \dot{N}$ はエージェント i と対応する。 $\dot{A}_w(t)$ はエージェント間での進路妨害を表すアークの有限集合であり、 $a_{jk} = (n_j, n_k) \in \dot{A}_w(t)$ 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。 $\dot{A}_w(t)$ は

$$\begin{aligned} \dot{A}_w(t) &= \sum_{j=1}^I \sum_{k=1}^I f_0(t, j, k), \\ f_0(t, j, k) &= \begin{cases} \{a_{jk}\} & \text{if } p_k^j(t) = 1 \\ \emptyset & \text{if } p_k^j(t) = 0 \end{cases} \end{aligned} \quad (14)$$

と求められる。 $\dot{G}_w(t)$ における n_j の出次数 $\text{outdeg}(n_j)$ と入次数 $\text{indeg}(n_j)$ は、それぞれエージェント j が受ける進路妨害とエージェント j による進路妨害の数を表す。有向グラフにおいて出次数、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。(9), (14)式より $\dot{G}_w(t)$ について

$$\text{outdeg}(n_j) \leq 1, \quad \forall n_j \in \dot{N} \quad (15)$$

が成立する。

文献 [6] ではエージェント間の進路妨害を表す有向グラフから出端点を削除することでデッドロックを検知し、出端点削除後のグラフから入端点を削除することでデッドロックの原因を特定している。ノードの削除は接続するすべてのアークの削除を意味する。本論文では $\dot{G}_w(t)$ から Algorithm 1 に従って入端点を削除した有向グラフ $\dot{G}_{wc}(t) = (\dot{N}_{wc}(t), \dot{A}_{wc}(t))$ をデッドロック検知・原因特定に用いる。 $|\dot{N}_{wc}(t)| = I$ より、Algorithm 1 では多くても 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。このとき、 $\dot{G}_{wc}(t)$ は

$$\text{indeg}(n_j) \geq 1, \quad \forall n_j \in \dot{N} \quad (16)$$

を満たす。(15), (16)式より $\dot{G}_{wc}(t)$ は

$$\text{outdeg}(n_j) = \text{indeg}(n_j) = 1, \quad \forall n_j \in \dot{N} \quad (17)$$

を満たす。(17)式より文献 [6] と同様に $\dot{G}_{wc}(t)$ は出端点を持たないオイラーグラフとなり、デッドロックの検知・

原因特定が達成される。

Algorithm 1 Identify agents causing deadlocks

```

1: Input  $G = (N_g, A_g)$ 
2:  $N'_g = N_g$ ;
3: while  $N'_g \neq \emptyset$  do
4:    $N'_g = \emptyset$ 
5:   for  $j = 1; j \leq I; j++$  do
6:     if  $n_j \in N_g$  &&  $\text{indeg}(n_j) == 0$  then
7:        $N'_g = N'_g \cup n_j$ ;
8:     end if
9:   end for
10:   $N_g = N_g \setminus N'_g$ ;
11: end while
12: return  $G$ 
  
```

デッドロックからの回復動作はエージェント間での探索ユニットの交換と疑似移動を用いた探索引継ぎで構成される。 $\dot{G}_{wc}^i(t)$ に対する探索ユニットの交換を

$$\mu^k(t+1) = \begin{cases} \mu^k(t) & \text{if } n_k \notin \dot{N}_{wc}(t) \\ \mu^j(t) & \text{if } a_{jk} \in \dot{A}_{wc}(t) \end{cases} \quad (18)$$

疑似移動動作を

$$\tilde{u}^k(t) = \begin{cases} \mathbf{0}_{|E|} & \text{if } n_k \notin \dot{N}_{wc}(t) \\ \hat{u}^j(t) & \text{if } a_{jk} \in \dot{A}_{wc}(t) \end{cases} \quad (19)$$

と与える。

4.2 Wait-for グラフの特徴

(2), (8), (14) 式より $\dot{G}_w(t)$ において

$$\begin{aligned} \text{outdeg}(n_i) = \text{outdeg}(n_j) = 1 &\implies \\ \delta_m(n_i, n_j) &\leq \delta_g(\dot{G}_w(t), n_i, n_j) \end{aligned} \quad (20)$$

が成り立つ。以降では、 $\dot{G}_{wc}(t)$ の連結成分を一つのデッドロックとして扱う。 $\dot{G}_{wc}(t)$ の連結成分を $\dot{G}_{wh}(t) = (\dot{N}_{wh}(t), \dot{A}_{wh}(t))$ と表す。 $\dot{G}_{wh}(t)$ について $|\dot{N}_{wh}(t)|$ をデッドロックの規模として扱う。 $\dot{G}_{wh}(t)$ はオイラー閉路であるため $\dot{G}_{wh}(t)$ 上の任意のノード n_i と n_j について

$$2 \delta_g(\dot{G}_{wh}(t), n_i, n_j) \leq |\dot{N}_{wh}(t)| \quad (21)$$

が成り立つ。

5. 通信手段

本章では通信可能範囲内の情報共有による (18), (19) 式の実現を目指す。探索引継ぎの表現にはエージェント i の引継ぎ対象の ID である $\sigma^i(t) \in \{1, 2, \dots, I\}$ と探索引継ぎ指令 $\hat{r}^i(t) \in \{0, 1\}^I$ を用いる。 $\hat{r}^i(t)$ の j 番目の要素を $\hat{r}_j^i(t)$ 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 の探索を引き継ぐ場合、 $\hat{r}_j^i(t)$ は 1 となる。そうでない場合、 $\hat{r}_j^i(t)$ は 0 となる。時刻 t における探索引継ぎを

$$\mu^i(t+1) = \mu^{\sigma^i(t)}(t), \quad (22)$$

$$\tilde{u}^i(t) = \begin{cases} \hat{u}^{\sigma^i(t)} & \text{if } \sigma^i(t) \neq \mu^i(t) \\ \mathbf{0}_{|E|} & \text{if } \sigma^i(t) = \mu^i(t) \end{cases} \quad (23)$$

と表す。 $\sigma^i(t)$ は探索引継ぎ指令 $\hat{r}^i(t)$ から

$$\sigma^i(t) = (1 - \hat{r}^i(t)^\top \hat{r}^i(t))i + [1 \ 2 \ \dots \ I] \hat{r}^i(t) \quad (24)$$

と与えられる。(24) 式より

$$a_{jk} \in \dot{A}_{wc}(t) \iff \hat{r}_j^k(t) = 1 \quad (25)$$

が成り立つとき (23) 式は (18), (19) 式を満たす。(25) 式において

$$a_{jk} \in \dot{A}_{wc}(t) \implies \hat{r}_j^k(t) = 1 \quad (26)$$

は発生したすべてのデッドロックに対してデッドロック解消のための探索引継ぎが行われることを意味する。同様に、

$$a_{jk} \in \dot{A}_{wc}(t) \iff \hat{r}_j^k(t) = 1 \quad (27)$$

はデッドロック解消以外の目的で探索引継ぎが行われないことを意味する。そこで、(26) 式が成り立たない場合をデッドロックの検知失敗、(27) 式が成り立たない場合をデッドロックの誤検知と表す。(25) 式の成立が保証される条件を検知保証条件と表す。

エージェントは通信可能範囲内の情報共有にもとづき $\dot{G}_w(t)$ および $\dot{G}_{wc}(t)$ の形状を予測し、 $\hat{r}^i(t)$ を決定する。エージェント i による $\dot{G}_w(t)$ と $\dot{G}_{wc}(t)$ の形状予測結果を、それぞれ $\hat{G}_w^i(t) = (\hat{N}_w^i(t), \hat{A}_w^i(t))$ 、 $\hat{G}_{wc}^i(t) = (\hat{N}_{wc}^i(t), \hat{A}_{wc}^i(t))$ と表す。

以降ではデッドロック解決のための通信手段として、シングルホップ型、マルチホップ型、共同検知型の三つの通信手段を示す。シングルホップ型はエージェントを中継した情報共有を用いない通信手段である。マルチホップ型は $\hat{G}_w^i(t)$ の作成と $\hat{G}_{wc}^i(t)$ の共有にエージェントを中継した情報共有を用いる通信手段である。共同検知型はエージェントが中継する情報の情報量を抑えた通信手段である。エージェントを中継した情報共有を行う場合の中継回数を $h \in \mathbb{N}$ と表す。

5.1 シングルホップ型

シングルホップ型では $\hat{A}_w^i(t)$ を

$$\hat{A}_w^i(t) = \sum_{j=1}^I \sum_{k=1}^I f_1(t, j, k), \quad (28)$$

$$f_1(t, j, k) = \begin{cases} \{a_{jk}\} & \text{if } \bar{p}_k^i(t, j) = 1 \\ \emptyset & \text{if } \bar{p}_k^i(t, j) = 0 \end{cases}$$

と求める。(13), (14), (28) 式より $\hat{G}_w^i(t)$ は

$$a_{jk} \in \hat{A}_w^i(t) \iff (a_{jk} \in \dot{A}_w(t) \wedge a_{ij} \in \dot{A}_w(t)) \quad (29)$$

を満たす。エージェント i は $\hat{G}_w^i(t)$ から Algorithm 1 に

従って入端点を削除し、得られたグラフを $\hat{G}_{wc}^i(t)$ とする。このとき、 $\hat{G}_{wc}^i(t)$ は $\hat{G}_w^i(t)$ に対して

$$\hat{G}_{wc}^i(t) \subseteq \hat{G}_w^i(t) \quad (30)$$

(29) 式より $\hat{G}_{wc}^i(t)$ に対して

$$\hat{G}_{wc}^i(t) \subseteq \hat{G}_{wc}^i(t) \quad (31)$$

を満たす。

$\hat{G}_{wc}^i(t)$ の共有には $\mathbf{R}^i(t) \in \{0,1\}^{I \times I}$ を用いる。 $\mathbf{R}^i(t)$ の j 列要素を $\mathbf{r}_j^i(t)$ と表し、 $\mathbf{r}_j^i(t)$ の k 番目の要素を $r_{j,k}^i(t)$ と表す。 $r_{j,k}^i(t)$ は

$$r_{j,k}^i(t) = \begin{cases} 1 & \text{if } a_{kj} \in \hat{A}_{wc}^i(t) \\ 0 & \text{otherwise} \end{cases} \quad (32)$$

と与えられる。(17), (31), (32) 式より $\mathbf{r}^i(t)$ は単位ベクトルか零ベクトルである。エージェントは $\hat{\mathbf{r}}^i(t)$ を

$$\begin{aligned} \hat{\mathbf{r}}^i(t, j) &= \xi(\bar{\mathbf{r}}_j^i(t, j), \hat{\mathbf{r}}^i(t, j-1)), \\ \xi(\lambda_1, \lambda_2) &= \lambda_1 \vee \lambda_2, \\ \hat{\mathbf{r}}^i(t, 0) &= \mathbf{0}_I, \\ \hat{\mathbf{r}}^i(t) &= \hat{\mathbf{r}}^i(t, I) \end{aligned} \quad (33)$$

と求める。 $\xi(\lambda_1, \lambda_2)$ は各要素が 0 か 1 である等しい長さの列ベクトル λ_1 と λ_2 の要素ごとの論理和を意味する。

このとき、(13), (29), (30), (32) 式より

$$\hat{r}_{j,k}^i(t, i) = 1 \iff a_{kj} \in \hat{A}_{wc}^i(t) \quad (34)$$

が成り立つ。(33), (34) 式より $\hat{\mathbf{r}}^i(t)$ は

$$\hat{r}_j^i(t) = 1 \iff a_{ji} \in \sum_{k=1}^I \hat{A}_{wc}^k(t) \quad (35)$$

を満たす。(31), (35) 式より、 $\hat{\mathbf{r}}^i(t)$ は (27) 式を満たす。また、(10)~(13), (21), (28), (35) 式より発生した $|\dot{N}_{wh}(t)| \leq 2d$ を満たすデッドロック $\dot{G}_{wh}(t)$ の検知が保証される。したがって、検知保証条件は

$$I \leq 2d \quad (36)$$

である。

5.2 マルチホップ型

$\hat{G}_w^i(t)$ の作成においてエージェント i の h_1 回目の送信情報を $\mathbf{P}_{h_1}^i(t) \in \{0,1\}^{I \times I}$ 、 h_1 回目の情報共有の結果を $\hat{\mathbf{P}}_{h_1}^i(t) \in \{0,1\}^{I \times I}$ と表す。 $\mathbf{P}_{h_1}^i(t)$ と $\hat{\mathbf{P}}_{h_1}^i(t)$ の j 列要素を、それぞれ $\mathbf{p}_{h_1,j}^i(t)$ 、 $\hat{\mathbf{p}}_{h_1,j}^i(t)$ と表す。 $i=j$ である場合 $\mathbf{p}_{h_1,j}^i(t)$ は $\mathbf{p}^i(t)$ に対応し、 $i \neq j$ である場合 $\mathbf{p}_{h_1,j}^i(t)$ はエージェント j から受け取った中継情報である。また、 $\mathbf{p}_{h_1,j}^i(t)$ と $\hat{\mathbf{p}}_{h_1,j}^i(t)$ の k 番目の要素を、それぞれ $p_{h_1,j,k}^i(t)$ 、 $\hat{p}_{h_1,j,k}^i(t)$ と表す。 $p_{h_1,j,k}^i(t)$ と $\hat{p}_{h_1,j,k}^i(t)$ はどちらも、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場合に 1 となる。そうでない場合、 $p_{h_1,j,k}^i(t)$ と $\hat{p}_{h_1,j,k}^i(t)$ は 0 となる。送信情報の初期値を

$$\mathbf{p}_{1,j}^i(t) = \begin{cases} \mathbf{p}^i(t) & \text{if } i=j \\ \mathbf{0}_I & \text{otherwise} \end{cases} \quad (37)$$

とする。情報共有の結果は、

$$\begin{aligned} \check{\mathbf{p}}_{h_1,j}^i(t, k) &= \xi(\check{\mathbf{p}}_{h_1,j}^i(t, k-1), \mathbf{p}_{h_1,j}^i(t, k)) \\ \check{\mathbf{p}}_{h_1,j}^i(t) &= \check{\mathbf{p}}_{h_1,j}^i(t, I) \\ \check{\mathbf{p}}_{h_1,j}^i(t, 0) &= \mathbf{p}_{h_1,j}^i(t) \end{aligned} \quad (38)$$

と求まる。送信情報の更新は

$$\mathbf{P}_{h_1+1}^i(t) = \hat{\mathbf{P}}_{h_1}^i(t) \quad (39)$$

と与えられる。(13), (37)~(39) 式より、 $\hat{\mathbf{P}}^i(t)$ について

$$\hat{\mathbf{p}}_{h_1,j}^i(t) = \begin{cases} \mathbf{p}^j(t) & \text{if } \delta_g(\dot{G}_n(t), n_i, n_j) \leq h_1 \\ \mathbf{0}_I & \text{if } \delta_g(\dot{G}_n(t), n_i, n_j) > h_1 \end{cases} \quad (40)$$

が成り立つ。また、(9), (39), (40) 式より $\mathbf{p}_{h_1,j}^i(t)$ は単位ベクトルか零ベクトルとなる。

マルチホップ型では $\hat{A}_w^i(t)$ を

$$\begin{aligned} \hat{A}_w^i(t) &= \sum_{j=1}^I \sum_{k=1}^I f_2(t, j, k), \\ f_2(t, j, k) &= \begin{cases} \{a_{jk}\} & \text{if } \hat{p}_{h,j,k}^i(t) = 1 \\ \emptyset & \text{if } \hat{p}_{h,j,k}^i(t) = 0 \end{cases} \end{aligned} \quad (41)$$

と求める。(13), (14), (40), (42) 式より $\hat{G}_w^i(t)$ は

$$\begin{aligned} a_{jk} \in \hat{A}_w^i(t) &\iff \\ &\left(a_{jk} \in \hat{A}_w(t) \wedge \delta_g(\dot{G}_n(t), n_i, n_j) \leq h \right) \end{aligned} \quad (42)$$

を満たす。

シングルホップ型と同様にエージェント i は $\hat{G}_w^i(t)$ から Algorithm 1 に従って入端点を削除し、得られたグラフを $\hat{G}_{wc}^i(t)$ とする。Algorithm 1 および (42) 式より、 $\hat{G}_{wc}^i(t)$ は (30), (31) 式を満たす。(30), (42) 式より $\delta_g(\dot{G}_n(t), n_i, n_j)$ は $\hat{G}_{wc}^i(t)$ について

$$\{n_i, n_j\} \subseteq \hat{N}_{wc}^i(t) \implies \delta_g(\dot{G}_n(t), n_i, n_j) \leq h \quad (43)$$

を満たす。

$\hat{G}_{wc}^i(t)$ の共有においてエージェント i の h_1 回目の送信情報を $\mathbf{R}_{h_1}^i(t) \in \{0,1\}^{I \times I}$ 、 h_1 回目の情報共有の結果を $\hat{\mathbf{R}}_{h_1}^i(t) \in \{0,1\}^{I \times I}$ と表す。 $\mathbf{R}_{h_1}^i(t)$ と $\hat{\mathbf{R}}_{h_1}^i(t)$ の j 列要素を、それぞれ $\mathbf{r}_{h_1,j}^i(t)$ 、 $\hat{\mathbf{r}}_{h_1,j}^i(t)$ と表す。 $\mathbf{r}_{h_1,j}^i(t)$ と $\hat{\mathbf{r}}_{h_1,j}^i(t)$ の k 番目の要素を、それぞれ $r_{h_1,j,k}^i(t)$ 、 $\hat{r}_{h_1,j,k}^i(t)$ と表す。送信情報の初期値を $\hat{G}_{wc}^i(t)$ から

$$\mathbf{r}_{1,j,k}^i(t) = \begin{cases} 1 & \text{if } a_{kj} \in \hat{A}_{wc}^i(t) \\ 0 & \text{otherwise} \end{cases} \quad (44)$$

と与える。情報共有の結果は、

$$\begin{aligned} \check{\mathbf{r}}_{h_1,j}^i(t, k) &= \xi(\check{\mathbf{r}}_{h_1,j}^i(t, k), \hat{\mathbf{r}}_{h_1,j}^i(t, k-1)) \\ \check{\mathbf{r}}_{h_1,j}^i(t) &= \check{\mathbf{r}}_{h_1,j}^i(t, I) \end{aligned} \quad (45)$$

$$\hat{\mathbf{r}}_{h_1,j}^i(t,0) = \mathbf{r}_{h_1,j}^i(t)$$

と求まる．送信情報の更新は

$$\mathbf{R}_{h_1+1}^i(t) = \hat{\mathbf{R}}_{h_1}^i(t) \quad (46)$$

と与えられる． $\hat{\mathbf{r}}^i(t)$ は

$$\hat{\mathbf{r}}^i(t) = \hat{\mathbf{r}}_{h,i}^i(t) \quad (47)$$

と求まる．(17), (31), (44)~(46) 式より $\hat{\mathbf{r}}_{h_1,j}^i(t)$ は単位ベクトルか零ベクトルとなる．

このとき，(13), (43)~(46) 式より $\mathbf{r}^i(t)$ は

$$\delta_g(\dot{G}_n(t), n_i, n_k) \leq h \implies \hat{\mathbf{r}}^i(t) \geq \mathbf{r}_{1,i}^k(t) \quad (48)$$

を満たす．また，(44)~(47) 式より

$$a_{ji} \in \sum_{k=1}^I \hat{A}_{wc}^k(t) \iff \hat{r}_j^i(t) = 1 \quad (49)$$

が成り立つ．(43), (44), (48), (49) 式より $\hat{\mathbf{r}}^i(t)$ は (35) 式を満たす．(31), (35) 式より，シングルホップ型と同様に $\hat{\mathbf{r}}^i(t)$ は (27) 式を満たす．(10)~(13), (35), (42) 式より発生した $|\dot{N}_{wh}(t)| \leq 2dh$ を満たすデッドロック $\dot{G}_{wh}(t)$ の検知が保証される．したがって，検知保証条件は

$$I \leq 2dh \quad (50)$$

である．

5.3 共同検知型

共同検知型では $\mathbf{p}^i(t)$ の共有によって $\dot{G}_w(t)$ の部分情報 $\hat{G}_w^i(t)$ を取得する． $\hat{G}_w^i(t)$ の形状から $\dot{G}_w(t)$ の入端点を特定し端点情報として共有することで， $\dot{G}_{wc}(t)$ の推定情報として $\hat{G}_{wc}^i(t)$ を求める．エージェントを中継した情報共有を端点情報の共有のみに適用するため，マルチホップ型と比較して共有情報の情報量を抑えることができる．

シングルホップ型と同様に $\hat{A}_w^i(t)$ を (28) 式によって求める．このとき， $\hat{G}_w^i(t)$ は (29) 式を満たす．(29) 式より $\hat{G}_w^i(t)$ について

$$\sum_{i=1}^I \hat{G}_w^i(t) = \dot{G}_w(t) \quad (51)$$

が成り立つ．

エージェント i は情報共有によって得た端点情報に従って $\hat{G}_w^i(t)$ から端点を削除し，得られた有向グラフにもとづいて自身の端点情報を更新する．端点情報の共有と更新を繰り返すことによって $\dot{G}_{wc}(t)$ を推定する．このとき，エージェント i の h_1 回目の送信情報を $m_{h_1}^i(t) \in \{0,1\}$ ， h_1 回目の情報共有によって得られる有向グラフを $\tilde{G}_{wc}^i(t, h_1) = (\tilde{N}_{wc}^i(t, h_1), \tilde{A}_{wc}^i(t, h_1))$ と表す． $\tilde{G}_{wc}^i(t, h_1)$ についてすべてのエージェントが持つ有向グラフを合わせたものを $\tilde{G}_{wc}(t, h_1) = (\tilde{N}_{wc}(t, h_1), \tilde{A}_{wc}(t, h_1))$ と表す． n_i が $\dot{N}_{wc}(t)$ に含まれる可能性がある場合 $m_{h_1}^i(t)$ は 1 となる．そうでない場合， $m_{h_1}^i(t)$ は 0 となる．

初期情報として $m_1^i(t)$ を

$$m_1^i(t) = \begin{cases} 1 & \text{if } \text{outdeg}(\hat{G}_w^i(t), n_i) = 1 \\ 0 & \text{otherwise} \end{cases} \quad (52)$$

と与える．エージェント i は共有情報をもとに $\tilde{N}_{wc}^i(t, h_1)$ を

$$\begin{aligned} \tilde{N}_{wc}^i(t, h_1) &= \sum_{j=1}^I f_3(t, j, h_1) \\ f_3(t, j, h_1) &= \begin{cases} \{n_j\} & \text{if } \bar{m}_{h_1}^i(t, j) = 1 \\ \emptyset & \text{otherwise} \end{cases} \end{aligned} \quad (53)$$

$\tilde{A}_{wc}^i(t, h_1)$ を

$$\begin{aligned} \tilde{A}_{wc}^i(t, h_1) &= \sum_{j=1}^I \sum_{k=1}^I f_4(t, j, k, h_1) \\ f_4(t, j, k, h_1) &= \begin{cases} \{a_{jk}\} & \text{if } a_{jk} \in \hat{A}_w^i(t) \\ & \wedge \bar{m}_{h_1}^i(t, j) = 1 \\ \emptyset & \text{otherwise} \end{cases} \end{aligned} \quad (54)$$

と求める．このとき， $\tilde{G}_{wc}(t, h_1)$ は

$$\tilde{G}_{wc}(t, h_1) = \sum_{i=1}^I \tilde{G}_{wc}^i(t, h_1) \quad (55)$$

である． $m_{h_1+1}^i(t)$ は $\tilde{G}_{wc}(t, h_1)$ から Algorithm 2 に従って求められる．Algorithm 2 は $m_{h_1}^i(t)$ について

$$m_{h_1}^i(t) = 0 \implies m_{h_1+1}^i(t) = 0 \quad (56)$$

を満たす．(29), (52), (56) 式より

$$\text{outdeg}(\dot{G}_w(t), n_i) = 0 \implies m_{h_1}^i(t) = 0 \quad (57)$$

が成り立つ．

$m_{h_1}^i(t)$ の更新は $\tilde{G}_{wc}(t, h_1)$ から d 回入端点の削除を行うことを意味する．Algorithm 2 においてエージェント i は $m_{h_1+1}^i(t)$ の更新のために $\tilde{G}_{wc}^i(t, h_1)$ において n_i の始点側の距離 $d-1$ の範囲に存在する n_j の入次数を参照する．このとき，(11), (13), (20), (28), (54), (57) 式より n_j について $\text{indeg}(\tilde{G}_{wc}^i(t, h_1), n_j) = \text{indeg}(\tilde{G}_{wc}(t, h_1), n_j)$ が成り立つため， $m_{h_1+1}^i(t)$ は $\tilde{G}_{wc}(t, h_1)$ から d 回入端点を削除した結果と一致する．また，(56) 式より $\tilde{G}_{wc}(t, h_1)$ における入端点削除の結果は $\tilde{G}_{wc}(t, h_1+1)$ に引き継がれる．したがって， $\tilde{G}_{wc}(t, h+1)$ は $\tilde{G}_{wc}(t, 1)$ から dh 回入端点を削除した結果と一致する．

このとき，(17), (52) 式より $m_1^i(t)$ は

$$m_1^i(t) = 0 \implies n_i \notin \dot{N}_{wc}(t) \quad (58)$$

を満たす．(58) 式より

$$\dot{G}_{wc}(t) \subseteq \tilde{G}_{wc}(t, 1) \subseteq \dot{G}_w(t) \quad (59)$$

が成り立つ．(59) 式より入端点の削除の結果得られる $\tilde{G}_{wc}(t, h+1)$ は

$$\dot{G}_{wc}(t) \subseteq \tilde{G}_{wc}(t, h+1) \quad (60)$$

を満たす.

Algorithm 2 Estimate agents causing deadlocks

```

1: Input  $\hat{G}_{wc}^i(t, h_1 - 1)$  as  $G = (N_g, A_g)$ 
2:  $N'_g = N_g$ ;
3: for  $h_2 = 1; h_2 \leq d; h_2++$  do
4:    $N'_g = \emptyset$ 
5:   for  $j = 1; j \leq I; j++$  do
6:     if  $n_j \in N_g$  &&  $\text{indeg}(n_j) == 0$  then
7:        $N'_g = N'_g \cup n_j$ ;
8:     end if
9:   end for
10:   $N_g = N_g \setminus N'_g$ ;
11: end for
12: if  $n_i \in N_g$  then
13:    $m_{h_1}^i(t) = 1$ ;
14: else
15:    $m_{h_1}^i(t) = 0$ ;
16: end if
17: return  $m_{h_1}^i(t)$ 
  
```

$\hat{r}^i(t)$ 決定のためエージェントは $m_{h+1}^i(t)$ を共有し, $\hat{G}_{wc}^i(t, h+1)$ を作成する. 共同検知型では $\hat{G}_{wc}^i(t, h+1)$ を $\hat{G}_{wc}^i(t)$ として扱う. $\hat{r}^i(t)$ は $\hat{G}_{wc}^i(t)$ から

$$\hat{r}_j^i(t) = \begin{cases} 1 & \text{if } a_{ji} \in \hat{A}_{wc}^i(t) \\ 0 & \text{otherwise} \end{cases} \quad (61)$$

と与えられる. (12), (29) 式より

$$(d \geq 1 \wedge a_{ji} \in \hat{A}_w(t)) \iff a_{ji} \in \hat{A}_w^i(t)$$

が成り立つこと, および (60) 式より (61) 式は (26) 式を満たす. $|\hat{N}| = I$ より $dh \geq I$ であれば端点削除が十分に行われるため $\hat{G}_{wc}^i(t) = \hat{G}_{wc}^i(t)$ となる. したがって, 検知保証条件は

$$I \leq dh \quad (62)$$

である.

5.4 通信負荷評価

通信手段ごとのデッドロック解決機能の性能を通信負荷と検知保証条件から比較する. デッドロック解決に必要な通信回数を各通信手段における通信負荷として扱う. 通信負荷は共有情報の情報量と情報共有の回数, 1 回の通信における情報量の上限 $\tau \in \mathbb{R}$ によって求まる. 制約 f) より共有情報の情報量が τ を超える場合, 共有情報を分割して通信する必要がある.

エージェントは通信に際して行列やベクトルで表される共有情報をスカラー値に変換する. $\mathbf{p}^i(t)$, $\mathbf{R}^i(t)$, $\mathbf{P}_{h_1}^i(t)$, $\mathbf{R}_{h_1}^i(t)$ のスカラー値表現をそれぞれ $\zeta(\mathbf{p}^i(t))$, $\zeta(\mathbf{R}^i(t))$, $\zeta(\mathbf{P}_{h_1}^i(t))$, $\zeta(\mathbf{R}_{h_1}^i(t))$ と表す. $\mathbf{p}^i(t)$, $\mathbf{r}_j^i(t)$, $\mathbf{p}_{h_1,j}^i(t)$, $\mathbf{r}_{h_1,j}^i(t)$ が単位ベクトルか零ベクトルであるこ

とから共有情報のスカラー値表現を

$$\begin{aligned} \zeta(\mathbf{p}^i(t)) &= \phi \mathbf{p}^i(t) \\ \zeta(\mathbf{R}^i(t)) &= \phi \mathbf{R}^i(t) \psi \\ \zeta(\mathbf{P}_{h_1}^i(t)) &= \phi \mathbf{P}_{h_1}^i(t) \psi \\ \zeta(\mathbf{R}_{h_1}^i(t)) &= \phi \mathbf{R}_{h_1}^i(t) \psi \\ \phi &= [1 \ 2 \ 3 \ \dots \ I] \\ \psi &= \begin{bmatrix} 1 & I_1 & I_1^2 & \dots & I_1^{I-1} \end{bmatrix}^T \\ I_1 &= I+1 \end{aligned} \quad (63)$$

と与える.

τ を考慮に入れた通信負荷を Table 1 に示す. Table 1 よりシングルホップ型は $h=1$ のマルチホップ型と比較してデッドロック解決に必要な共有情報の情報量が少ないことがわかる. このとき, それぞれの通信手段に対する検知保証条件 (36) 式と (50) 式が一致する. また, 共同検知型では $m_{h_1}^i(t)$ の情報量が 1bit であるため, 他の通信手段と比較して共有情報の情報量が抑えられる. (50), (62) 式および Table 1 より共同検知型は $\tau < \lceil I \log_2(I+1) \rceil$ の場合において, マルチホップ型に比べて検知保証条件達成に要する通信負荷を削減できる.

マルチホップ型の長所として, 求めるデッドロック検知性能に応じた h と通信負荷の調整が可能な点が挙げられる. マルチホップ型では検知保証条件を満たさない場合にも h に対して $|\hat{N}_{wh}(t)| \leq 2dh$ の規模のデッドロックの検知が保証される. 一方で, 共同検知型では検知保証条件を満たさない場合, デッドロックの規模によらず探索ユニットの消失や重複割り当てが発生する可能性がある. 大規模なデッドロックの発生頻度は迷路に対するエージェントの密度や経路計画手法によって変化するため, マルチホップ型では予想されるデッドロックの最大規模に応じて h を変化させることでデッドロックの検知失敗が発生しない範囲で通信負荷を抑えることができる.

Table 1 Communication load

Communication protocol	Communication load
Single hop	$\frac{\lceil \log_2(I+1) \rceil}{\tau} + \frac{\lceil I \log_2(I+1) \rceil}{\tau}$
Multi hop	$2h \frac{\lceil I \log_2(I+1) \rceil}{\tau}$
Cooperative detection	$\frac{\lceil \log_2(I+1) \rceil}{\tau} + h+1$

各通信手段の得意とする状況の違いは, 情報の伝達経路の違いによるものである. シングルホップ型とマルチホップ型では共有情報にもとづいて各エージェントが独立してデッドロックを検知し, その結果を周囲のエージェントと共有する. 情報の伝達経路は $\hat{G}_n(t)$ に沿った

ものであり、1回の通信における伝達距離は $\hat{G}_n(t)$ 上で1である。共同検知型では各エージェントは情報共有によって部分情報 $\hat{G}_w^i(t)$ もとづいて $\tilde{G}_{wc}(t, h_1)$ 上の端点を削除する。このとき、端点情報の変更の順序は $\tilde{G}_{wc}(t, 1)$ に従う。端点情報の変更を情報の伝達と捉えると伝達経路は $\tilde{G}_{wc}(t, 1)$ に沿ったものであり、1回の通信における伝達距離は $\tilde{G}_{wc}(t, 1)$ 上で最大 d である。共同検知型では $\mathbf{p}^i(t)$ の共有と $\hat{G}_w^i(t)$ の作成によって情報の伝達経路を形成し、伝達経路を付加情報として扱うことで共有情報の情報量を削減している。また、エージェントを中継した情報共有の利用方法の違いが検知保証条件を満たさない場合の(26), (27)式の達成状況に影響している。シングルホップ型とマルチホップ型において dh は情報共有の範囲を表すのに対し、共同検知型において dh は端点削除の回数を表す。そのため、シングルホップ型とマルチホップ型では(27)式の成立が保証され、共同検知型では(26)式の成立が保証される。(26), (27)式より、共同検知型ではシングルホップ型とマルチホップ型と比較して疑似移動の発生頻度が高くなる。

6. 探索効率評価

6.1 流量の予測

本論文では迷路探索の効率評価に流量を用いる。流量は区画からエージェントが移動する確率を表す。各区画からエージェントが区画移動を行う確率の平均値と区画移動または疑似移動を行う確率の平均値を、それぞれエージェント流量、ユニット流量と表す。探索終了までのエージェント流量とユニット流量の時間平均を、それぞれ $\bar{Q}$, $\bar{\dot{Q}}$ と表す。流量は探索におけるエージェントの移動の活発さを表し、探索対象上のエージェントの密度に対して流量が高いほど文献[6]のエージェントの稼働率が高くなる。探索期間が T と与えられるとき、 $\bar{Q}$ と $\bar{\dot{Q}}$ はそれぞれ

$$\bar{Q} = \frac{1}{T|V|} \sum_{t=1}^T \sum_{i=1}^I \mathbf{u}^i(t)^T \mathbf{1}_{|E|} \quad (64)$$

$$\bar{\dot{Q}} = \frac{1}{T|V|} \sum_{t=1}^T \sum_{i=1}^I (\mathbf{u}^i(t) + \tilde{\mathbf{u}}^i(t))^T \mathbf{1}_{|E|} \quad (65)$$

と与えられる。ただし、 T については制約a)を考慮し、エージェントの移動と通信の繰り返し回数を時間の指標とする。

流量の評価基準にはエージェントの移動経路と類似形状を持つASEPの直列ネットワークにおける流量を用いる。ASEPはエージェントの移動を同期するパラレルアップデート型と移動を同期しないランダムアップデート型に分けられる。本論文では制約a)を考慮し、ランダムアップデート型ASEPの直列ネットワークの流量解析[13]をパラレルアップデート型ASEPの直列ネットワークに拡張する。

ASEPは1次元の格子構造をしており、エージェントは始端から終端に向かって一定の確率で1区画ずつ移動

する。ASEP上での移動では、すでに他のエージェントが存在する区画への移動が制限される。本論文では移動確率を1とし、衝突回避のための停止を除いてエージェントの移動は必ず行われるものとする。ASEPネットワークの例をFig. 2に示す。ASEPネットワークはASEPとASEPの始端と終端に接続する頂点サイトで構成される。ASEPの終端に到着したエージェントはASEPの終端に接続している頂点サイトに移動する。頂点サイトに存在するエージェントは頂点サイトに始端が接続するASEPから等確率で一つを選択し、選択したASEPの始端に移動する。複数のエージェントが同一の頂点サイトに移動を試みた場合、等確率で選択された一つのエージェントのみが頂点サイトに移動する。また、他のエージェントが存在する頂点サイトへの移動は制限される。これにより、制約b)を表現する。

Fig. 2 ASEP network

ASEPネットワークにおける流量は頂点サイトの次数とエージェントの密度によって求まる。以降ではASEPネットワークにおける頂点サイトの次数は一定であるものと仮定する。頂点サイトの入次数を c 、ASEPにおけるエージェントの密度を ρ と表す。また、ASEPにおける流量を c と ρ に対して $\hat{Q}(c, \rho)$ と表す。頂点サイトにおける流量を c と ρ に対して $\hat{Q}_v(c, \rho)$ と表す。

文献[12]によると単体のASEPについてエージェントの密度が一定である場合、流量は密度に依存する。単体のASEPにおける流量を ρ に対して $\hat{Q}_e(\rho)$ と表す。文献[12]より、 $\hat{Q}_e(\rho)$ は

$$\hat{Q}_e(\rho) = \frac{1 - \sqrt{1 - 4\rho(1 - \rho)}}{2} \quad (66)$$

と与えられる。(66)式は

$$\hat{Q}_e(\rho) = \begin{cases} \rho & \text{if } 0 \leq \rho \leq \frac{1}{2} \\ 1 - \rho & \text{if } \frac{1}{2} < \rho \leq 1 \end{cases} \quad (67)$$

と変換できる。(67)式より $\rho = \frac{1}{2}$ のとき $\hat{Q}_e(\rho)$ は最大値 $\frac{1}{2}$ をとる。

ASEPネットワークでは頂点サイトにおいて接続するASEPの流れの合流・分離が発生するため、ASEPの接続数に応じて頂点サイトの密度と流量が増加する。したがって、 $\hat{Q}_v(c, \rho)$ は $\hat{Q}(c, \rho)$ に対して

$$\hat{Q}_v(c, \rho) = c\hat{Q}(c, \rho) \quad (68)$$

と与えられる。ここで、頂点サイトを区画数1のASEP

と考える. ASEP における流量の最大値が $\frac{1}{2}$ であることを考慮すると, (67), (68) 式より $\hat{Q}_v(c, \rho)$ は

$$\hat{Q}_v(c, \rho) = \begin{cases} \frac{1}{2} & \text{if } \frac{1}{2c} < \rho < 1 - \frac{1}{2c} \\ c \frac{1 - \sqrt{1 - 4\rho(1 - \rho)}}{2} & \text{otherwise} \end{cases} \quad (69)$$

$\hat{Q}(c, \rho)$ は

$$\hat{Q}(c, \rho) = \begin{cases} \frac{1}{2c} & \text{if } \frac{1}{2c} < \rho < 1 - \frac{1}{2c} \\ \frac{1 - \sqrt{1 - 4\rho(1 - \rho)}}{2} & \text{otherwise} \end{cases} \quad (70)$$

と求められる. (69), (70) 式は頂点サイトにおける流量が上限に達した場合, 接続する ASEP の流量が制限されることを意味する.

エージェントの移動経路の ASEP ネットワークへの変換例を Fig. 3 に示す.

Fig. 3 ASEP network expression of the exploration path

Fig. 3 (b) の頂点サイトは (a) における移動経路の始端と終端, および移動方向が複数存在する区画に対応する. (b) の ASEP は (a) においてエージェントの移動方向が 1 方向に限定される区画群に対応する. エージェントの移動経路の ASEP ネットワークへの変換によって, エージェントが使用する経路計画手法と探索対象の形状に応じた流量予測が可能となる.

6.2 数値実験

デッドロック解決のための通信手段が流量に与える影響を数値実験により検証する. 本実験では通信可能範囲を $d=1$ に設定する. デッドロック検知能力が同等になるよう, $h=1$ のマルチホップ型 (シングルホップ型) と $h=2$ の共同検知型, $h=5$ のマルチホップ型と $h=10$ の共同検知型の探索結果を比較する. 探索期間を $T=100$ に設定し, エージェント数を 1, 10, 20, 30, ..., 80 と変化させる. すべての探索ユニットはミッションを持たず, 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. 移動経路はエージェントの初期位置と迷路上の障害物の位置を考慮したものであり, 障害物回避のほかに移動方向の制約を持たない. 探索対象となる迷路を Fig. 4 に示す.

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. 4 の灰色の区画は行き止まりに相当する区画であり, 経路計画手法によらず移動方向が 1 方向に限定される. Fig. 4 について障害物が存在しないアークすべてがエージェントの移動経路に含まれると仮定する. このとき, 行き止まりとなる区画が ASEP に変換されるため, エージェントの移動経路の ASEP ネットワーク表現における頂点サイトの数は 85 となる. 頂点サイトに対応する区画群に隣接する通過可能なアーク数が 217 であることから, エージェントの移動経路の ASEP ネットワーク表現における平均次数は $c=2.55$ となる. そこで, 流量の評価指標を $\hat{Q}(2.55, \rho)$ に設定する. 本実験ではエージェントの移動経路による影響を考慮し, 同一のエージェント数に対してエージェントの初期位置とエージェントの移動経路を変更し 5 回の実験を行った.

Fig. 4 Simulation layout

5 回の探索のエージェント流量とユニット流量の平均値をそれぞれ Fig. 5 と Fig. 6 に示す. Fig. 5 より通信手段によらずエージェント流量は予測流量よりも低くなるのがわかる. これはデッドロックの発生そのものが流量の低下を引き起こすためである. Fig. 6 の $h=5$ のマルチホップ型と $h=10$ の共同検知型の結果に着目すると, Fig. 5 で確認された流量の低下が改善され予測流量に近づいていることが確認できる. このことから, 提案手法によるデッドロック解決では探索引継ぎのための疑似移動によってデッドロックによる流量低下が抑制されているといえる. $\rho \leq 0.2$ の範囲では $h=5$ のマルチホップ型と $h=10$ の共同検知型について大きな流量の差が見られなかった. これは, 検知保証条件成立下での探索引継ぎが (25) 式に固定されるためである.

また, Figs. 5, 6 よりマルチホップ型では $h=1$ とした場合に $h=5$ の場合と比較してエージェント流量, ユニット流量共に低くなることが確認された. これは, デッドロックの検知失敗により, 一部のエージェントが膠着状態に陥ったためである. このことから, マルチホップ型

の通信手段を用いる場合デッドロック検知能力の不足は流量の低下を引き起こすといえる。一方で、共同検知型では $h=2$ とした場合に $h=10$ の場合と比較してユニット流量が高くなっていることが確認された。流量の増加傾向は ρ が高くなるほど強くなり、予測流量を大きく上回る流量が確認された。これは、デッドロックの誤検知により、疑似移動の頻度が上昇したためである。このことから、共同検知型の通信手段を用いる場合デッドロック検知能力の不足は流量の増加に貢献するといえる。ただし、デッドロックの誤検知に起因する疑似移動では、役割交換が正常に行われずミッションの達成が不可能になる状況が想定される。そのため、通信手段の選択ではエージェントに与える役割を考慮する必要がある。

Fig. 5 Simulation result (Q) Fig. 6 Simulation result ($\hat{Q}$)

7. おわりに

分散協調制御型探索システムによる迷路の協調探索における、デッドロック解決のための通信手段としてシングルホップ型、マルチホップ型、共同検知型の三つを提案した。提案した通信手段の比較を通じて、通信手段が迷路探索の効率に与える影響を考察した。提案する三つの通信手段はそれぞれ共有情報と情報の伝達経路、デッドロック検知能力不足による探索への影響が異なる。提案する通信手段についてエージェントの通信機能の性能に対するデッドロック検知能力と通信負荷を導出した。マルチホップ型はエージェントを中継した情報共有を用いており、中継数を増やすことでデッドロック検知能力

を向上させることが可能である。共同検知型は中継情報の導入による共有情報の情報量増加の対策として、共有情報の情報量を抑えた通信手段である。探索効率への影響評価では探索対象の形状とエージェントの移動経路に対するエージェントの密度と流量の関係を解析し、通信可能範囲が不足する状況において通信手段が流量に与える影響を数値実験により検証した。以上の結果から、分散協調探索システムによる迷路の協調探索における通信手段の設定指針について議論した。

今後の課題としてエージェントへの役割の割り当て状況を考慮した探索効率評価が挙げられる。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] T. R. Wanasinghe and K. I. Mann: Distributed leader-assistive localization method for a heterogeneous multirobotic system; *IEEE Transactions on Automation Science and Engineering*, Vol. 12, No. 3, pp. 795–809 (2015)
- [2] Y. Zhang, F. Bastani, I. L. Yen, J. Fu and I. R. Chen: Availability analysis of robotic swarm systems; *14th IEEE Pacific Rim International Symposium on Dependable Computing*, pp. 331–338 (2008)
- [3] K. Knight: Are many reactive agents better than a few deliberative ones?; *Proceedings of the Thirteenth International Joint Conference on Artificial Intelligence*, pp. 432–437 (1993)
- [4] 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)
- [5] 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)
- [6] 望月，澤田，新：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；システム制御情報学会論文誌, Vol 31, No 4, pp. 167–176 (2018)
- [7] N. Wu and M. C. Zhou: AGV routing for conflict resolution in AGV system; *International Conference on Robotics & Automation*, pp. 14–19 (2003)
- [8] L. Kalinovic, 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)
- [9] E. H. Kivelevich and K. Cohen: Multi-agent maze exploration; *Journal of Aerospace Computing, Information, and Communication*, Vol. 7, pp. 391–405 (2010)
- [10] S. A. Reveliotis: Conflict resolution in AGV systems; *IIE Transactions*, Vol. 32, pp. 647–659 (2000)
- [11] 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)

- [12] A. Schadschneider and M. Schreckenberg: Cellular automaton models and traffic flow; *Journal of Physics A: Mathematical and Theoretical*, Vol. 26, No. 15, pp. 1–7 (1993)
- [13] 江崎, 今村: ネットワーク上の非対称単純排他過程 (ASEP); 応用数理, Vol. 23, No. 5, pp. 7–14 (2013)

著者略歴

もちづき ゆかり
望月優加里

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, 2018年3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修了。

さわだ 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。2016年より制御システムセキュリティセンター顧問。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。

しん 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。2012年より制御システムセキュリティセンター理事長。制御理論を中心に広く工学全体に興味を持つ。1991年, 1993年, 1998年計測自動制御学会論文賞, 1992年同賞武田賞, 2006年同技術賞受賞。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。