

仮想グリッドネットワークにおける 経路最適化分散アルゴリズムの改良

宮川 歩^{1,a)} 金 鎔煥^{1,b)} 片山 喜章^{1,c)}

概要: 本研究では、仮想グリッドネットワークにおける最適ルーティング経路を自律的に生成する分散アルゴリズムを紹介しその改良を行う。仮想グリッドネットワークとは、無線ネットワークがカバーしている領域を仮想的に正方形のグリッドセル分割し、各セル毎に一つの代表ノード（ルータ）を決定し、隣接しているセルのルータ同士を連結させた格子状の仮想ネットワークである。仮想グリッドネットワークを用いたルーティングは、エネルギー効率の良さとトポロジの単純化からよく利用されている。仮想グリッドネットワークでは、ターゲット（宛先のノード）が異なるグリッドセルに移動した場合、現在保持しているルーティング経路をそのグリッドセルまで延長させることで、ターゲットまでの経路を容易に維持させることができる。しかし、このような経路の更新は、経路の延長だけを使っているため、ターゲットが移動を繰り返した場合、ネットワーク上のルーティング経路が冗長になってしまう恐れがある。このような冗長な経路は、エネルギー効率の悪化などの原因となるため、グリッドネットワーク上のルーティング経路を最短化する手法が必要である。このような問題を各ルータが持つ局所情報だけで解決する既存研究が既に提案されているが、本稿では、既存研究を改良し、より少ない局所情報だけで同じ問題（経路最適化）を解決するアルゴリズムを提案する。本研究により、各ノードはより少ない情報だけを利用するため、必要メモリ量を削減でき、さらに通信コストも削減することが期待できる。

AYUMU MIYAKAWA^{1,a)} YONGHWAN KIM^{1,b)} YOSHIKI KATAYAMA^{1,c)}

Abstract: In this paper, we introduce a local-information-based self-optimizing routing protocol in virtual grid networks and we propose an improvement of the previous routing algorithm. A virtual grid network is obtained by virtually dividing a wireless network into a grid of geographical square regions called cells, a specific node is elected as a router on each cell and each router can communicate with routers on adjacent cells. We consider a routing path between a single source node and a single target node. When a target node moves to a cell next to the current one, the routing path is extended to the next cell. However, the repetition of these extension processes may cause a redundant routing path, therefore a optimizing process is required to shorten the redundant routing path to save energy consumption. A local-information-based self-optimizing routing protocol is proposed previously, but we proposed a new self-optimizing routing protocol using less information. Our proposed algorithm can resolve the same problem using a smaller memory and promise higher communication efficiency for sharing local information.

1. はじめに

1.1 研究の背景

無線端末（以降、ノードとする）の普及により、ノードを用いた無線ネットワークが注目されている [7-10]。その例としてモバイルアドホックネットワーク (MANET)[2][3]

や無線センサネットワーク (WSNs)[1][2] がある。モバイルアドホックネットワーク (MANET) とはノード間の通信に基地局を用いず、ノードのみでネットワークを構築するものである。無線センサネットワーク (WSNs) とは、無線通信可能な多数のセンサを通信可能範囲を考慮して配置し、センサが取得した情報をセンサ同士の通信によって基地局に伝搬するものである。各ノードの通信範囲は限られており、この通信範囲内に存在するノードとのみ直接通信可能である。そのため、通信先のノード（以降、ターゲットとする）が通信元のノード（以降、ソースノードとする）

¹ 名古屋工業大学大学院
〒466-8555 愛知県名古屋市昭和区御器所町

a) miyakawa@moss.elcom.nitech.ac.jp

b) kim@nitech.ac.jp

c) katayama@nitech.ac.jp

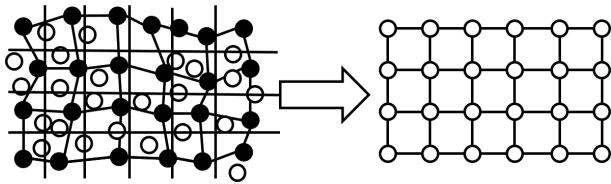


図 1 仮想グリッドネットワークの構成

通信範囲外に存在する場合には、他のノードを中継してマルチホップ通信を行う。マルチホップ通信ではターゲットまでの通信経路を構築する（以降、ルーティングとする。）必要がある。これまでに多数のルーティングプロトコルが提案されている [1][4-6][11-17]。それらの違いは通信経路の評価尺度であり、評価尺度にはホップ数、通信遅延、消費エネルギー、故障耐性などがある [1]。中でも優れているとされる評価尺度はホップ数であり、ホップ数を減らすことにより通信遅延や消費エネルギーを減らすことが期待される。そのためもっともよく利用されている評価尺度の一つである。本稿では、無線ネットワークのルーティングにおいて、効率よく最短経路を生成する自立分散アルゴリズムを提案する。

1.2 関連研究

1.2.1 仮想グリッドネットワーク

本稿では、仮想グリッドネットワーク上での通信経路を扱う。仮想グリッドネットワーク [6] は図 1 に示すように無線ネットワークを仮想的に正方形のセルに格子状に分割することにより得られる。このとき、各セルに 1 つ以上のノードが存在するように分割する。セルのサイズは同じセルにあるノードと隣り合うセルにあるノードが直接通信できるように決められる。例えば、すべてのノードの通信半径が R 以上のとき、すべてのセルの一边の大きさは $\frac{R}{\sqrt{5}}$ 以下となるように決定する。

無線ネットワークを分割した後、セルの内部から代表のノードを（以降、ルータとする）選ぶ。その後、セルが辺を共有して隣接するとセルのルータ同士が通信リンクをつなぐことによってネットワークを形成する。

セルの内部にあるルータに選ばれていないノードはセルの内部の通信において関係を持たないので、エネルギー消費を節約するためにスリープ状態にすることができる。そのため、エネルギー消費において優れた手法である。エネルギーの効率性のため、各セルのルータは定期的に再選択される。

仮想グリッドネットワーク上でのルーティングを行う手法として Geographic Adaptive Fidelity (GAF) が提案されている [6]。Geographic Adaptive Fidelity (GAF) とは、エネルギーが多く残っているノードがグリッドの代表ノードとなり、そのルータがセルの中継を行うという手法である。仮想グリッドネットワーク上でデータを通信するとき、

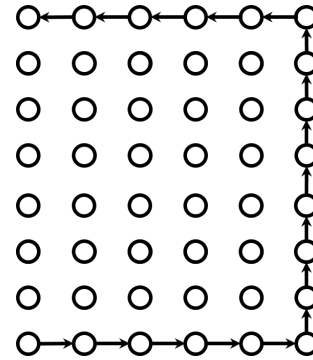


図 2 冗長な経路

メッセージのソースノードは初めにそのノードが属しているセルのルータにデータを送られる。そして、ルータを中継してターゲットの属しているセルまで通信が行われ、最後にターゲットの属しているセルのルータからターゲットにデータが送られる。

1.2.2 経路最適化問題

冗長な経路を短くする問題は経路最適化問題として分類される。

ターゲットの位置を認識できるルーティングプロトコルによってゼロから最短の経路を作ることで、経路最適化問題が解決することは簡単である。しかし、大規模な無線ネットワークではターゲットの正確な位置を検知するが困難な場合がある。

従って本研究では、ネットワークの属するすべてのノードが自分の回りの局所情報だけで経路全体を最適にする自立分散アルゴリズムを提案する。各ノードは局所情報しか利用しないため、大量のメモリや計算能力を必要とせず、ネットワーク規模に依存せず経路最適化問題を解決することが可能となる。

しかし、図 2 のような冗長な経路の場合システム全体の情報なしに経路の冗長な部分を見つけることは困難である。このように、すべての与えられた経路において最短経路を生成することを保証する局所的情報に基づいたルーティングプロトコルを設計することは難しい問題である。

1.2.3 Zigzag アルゴリズム

仮想グリッドネットワーク上での経路最適化問題を解く手法として Zigzag アルゴリズムが提案されている [18]。Zigzag アルゴリズムとは、経路の更新を繰り返すことによって経路を最短経路に収束させるアルゴリズムである。仮想グリッドネットワーク上のある経路 P が与えられたとき、以下の更新を繰り返し経路に適用することによって経路を最短経路に収束させる。 [18]

shortcut1 $\vec{p}_{i-1} \vec{p}_i \cdot \vec{p}_i \vec{p}_{i+1} = -1$ のとき、 (p_{i-1}, p_i, p_{i+1}) を p_{i-1} に置き換える。(図 3)

shortcut2 $\vec{p}_{i-1} \vec{p}_i \cdot \vec{p}_i \vec{p}_{i+1} = 0$ and $\vec{p}_{i-1} \vec{p}_i \cdot \vec{p}_{i+1} \vec{p}_{i+2} = -1$ のとき、 $(p_{i-1}, p_i, p_{i+1}, p_{i+2})$ を (p_{i-1}, p_{i+2}) に置き換える。(図 4)

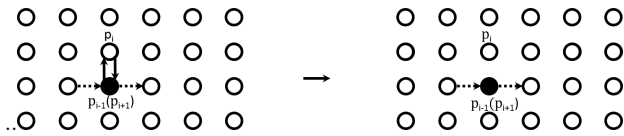


図 3 shortcut1

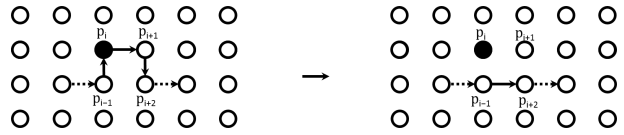


図 4 shortcut2

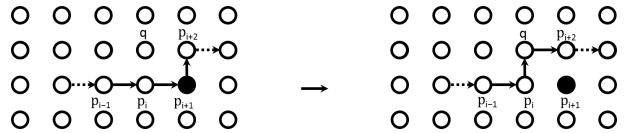


図 5 zigzag

zigzag $\overrightarrow{p_{i-1}p_i} \cdot \overrightarrow{p_i p_{i+1}} = 1$ and $\overrightarrow{p_i p_{i+1}} \cdot \overrightarrow{p_{i+1} p_{i+2}} = 0$ のとき, (p_i, p_{i+1}, p_{i+2}) を (p_i, q, p_{i+2}) に置き換える. q は $(p_i, q) = (p_{i+1}, p_{i+2})$ を満たすルーターとする. (図 5)

Zigzag アルゴリズムは, 各ルーターは経路上の前後 3 ホップ先までのルーティング情報を把握していることを仮定した上で正しい動作を保証している. 本研究では, 既存研究である Zigzag アルゴリズムと同様のモデルと問題を, さらに少ない情報 (経路上の前後 2 ホップ先まで) だけを利用して解決する自律分散アルゴリズムを提案する.

1.3 本研究の貢献

提案アルゴリズムでは, 関連研究の Zigzag アルゴリズムの視野を経路上 2 ホップの距離に限定している. 提案アルゴリズムの特徴として以下の特徴が挙げられている. [18]

- 1) 提案アルゴリズムは, 2 ホップの視野で Zigzag プロトコルと同じ経路最適化問題を解決することが出来る.
- 2) 提案アルゴリズムは, Zigzag プロトコルより情報を保存するメモリを少なくすることができ, 情報を取得する際の通信少なくすることが出来る.
- 3) 提案アルゴリズムは, 既存研究の Zigzag プロトコルと同じ環境で実行することが出来る.

1.4 論文の構成

本論文の構成は次の通りである. 第 2 章では提案アルゴリズムのモデルを示す. 第 3 章では提案アルゴリズムの実現法を示す. 第 4 章では提案アルゴリズムの正当性を示す. 第 5 章では提案アルゴリズムの評価実験を示す. 最後に第 6 章でまとめを述べる.

2. 提案アルゴリズムの実現法

2.1 仮想グリッドネットワークのモデル

仮想グリッドネットワークは無線ネットワークをセルと呼ばれる正方形のグリッド状に分割することによって得ることが出来る. 通信リンクの集合を V_g , ルーターの集合を E_g , 仮想グリッドネットワークを N_g と表すと, 仮想グリッドネットワークは $N_g = (V_g, E_g)$ の無向辺グラフと表される. 各ルーターは固有のグリッド識別子 id_i を保有している. ルーター $p_i, p_j \in V_g$ がそれぞれを含むセルが辺を共有して隣接しているときのみ通信リンク $(p_i, p_j) \in E_g$ が存在する. これにより, ルーターが仮想的にグリッド状に配置されているとみなすことができ, 仮想グリッドネットワーク N_g は格子グラフとなる. よって, $(p_i, p_j), (p_i, p_j) \in E_g$ であるとき, $\overrightarrow{p_i p_j} \cdot \overrightarrow{p_j p_k} = 0$ または $|\overrightarrow{p_i p_j} \cdot \overrightarrow{p_j p_k}| = 1$ が成り立つものとする.

ルーター同士の通信は非同期通信である. 各ルーターは位置情報を持たないものとする. ルーターは経路全体に関する情報を持っておらず, 経路上 2 ホップの経路情報のみを持っている. 経路情報はルーターは方向ラベル U (Up), D (Down), R (Right), L (Left) を通信することによって経路の状況をやり取りしている. (図 6) 経路上の各ルーターに方向ラベルを保存するよってルーターは経路情報を把握している.

2.2 仮想グリッドネットワーク上での経路最適化問題

仮想グリッドネットワーク上で, ターゲットが現在のセルから隣接するセルに移動したとき, 仮想グリッドネットワーク上の経路は今までの通信経路をターゲットが移動したセルに伸ばすことによって行われる. しかし, このような更新を繰り返すと経路が冗長になる場合がある. 今回提案するアルゴリズムは, このようにして生じた冗長な通信経路を仮想グリッドネットワークをホップ数の最小の通信経路に更新することを考える. 各ルーターは経路全体の情報を持っていない, 経路上の各ルーターに分散して保存されている.

経路上の各ルーターに保存されている経路情報は 2 種類あり, 経路情報とルーティング情報の 2 種類である.

経路情報は各通信リンクの情報であり, 自身の直前と直後の通新リンクの情報が方向ラベルを保存することによって保管される. 経路を更新する際に経路情報も更新される.

ルーティング情報は経路を更新するとき使用する情報である. 視野の範囲の経路情報が方向ラベルとして保管される. 経路情報とルーティング情報は同じルーターを複数回通る場合, その回数分ルーターに用意される. そのため, 経路が複数回同じルーターを通る場合に経路情報とルーティング情報は何回目に通る情報なのかを区別することが出来る.

例えば, 図 7 に示す経路の場合, p_2 における経路情報

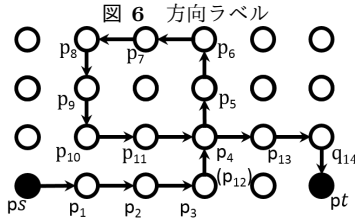
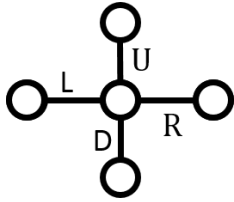


図 7 通信経路の例

	(pre_1, suc_1)	(pre_2, suc_2)
p_2	(L,R)	-
p_4	(D,U)	(D,U)

表 1 経路情報

	$(pre'_1, pre_1, suc_1, suc'_1)$	(pre_2, suc_2)
p_2	(L,L,R,U)	-
p_4	(L,D,U,U)	(L,L,R,U)

表 2 ルーティング情報

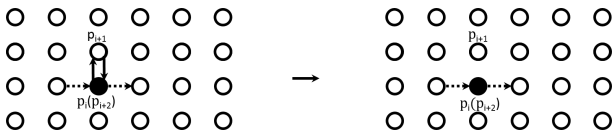


図 8 shortcut1

は経路が p_2 を 1 度通っているのので 1 つ用意され、 $(pre_1, suc_1)=(L, R)$ となる。 p_4 における経路情報は、経路が p_4 を 2 度通っているのので 2 つ用意され、 $(pre_1, suc_1)=(D, U)$ 、 $(pre_2, suc_2)=(L, R)$ となる。(表 1)

ルーティング情報に関しては、 p_2 における経路情報は経路が p_2 を 1 度通っているのので 1 つ用意され、 $(pre'_1, pre_1, suc_1, suc'_1)=(L, L, R, U)$ となる。 p_4 における経路情報は、経路が p_4 を 2 度通っているのので 2 つ用意され、 $(pre'_1, pre_1, suc_1, suc'_1)=(L, D, U, U)$ 、 $(pre'_2, pre_2, suc_2, suc'_2)=(L, R)$ となる。(表 2)

2.3 局所的な経路更新

提案アルゴリズムは局所的な更新を繰り返すことにより与えられた経路を最短経路に収束させる。その局所的な更新は次のように表される。

shortcut1 $\overrightarrow{p_i p_{i+1}} \cdot \overrightarrow{p_{i+1} p_{i+2}} = -1$ のとき、 (p_i, p_{i+1}, p_{i+2}) を p_i に置き換える。(図 8)

shortcut2 $\overrightarrow{p_{i-1} p_i} \cdot \overrightarrow{p_i p_{i+1}} = 0$ and $\overrightarrow{p_{i-1} p_i} \cdot \overrightarrow{p_{i+1} p_{i+2}} = -1$ のとき、 $(p_{i-1}, p_i, p_{i+1}, p_{i+2})$ を $(p_{i-1}, p_i, p_{i-1}, p_{i+2})$ に置き換える。(図 9)



図 9 shortcut2

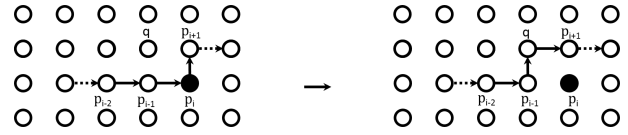


図 10 zigzag

p_i での局所的な更新		shortcut1	shortcut2	zigzag
変化する経路	p_{i-1}	-	-	suc
	p_i	suc	suc	pre, suc
	p_{i+1}	pre, suc	pre, suc	pre
	p_{i+2}	pre	pre	-
	q	-	-	pre, suc

表 3 経路情報

zigzag $\overrightarrow{p_{i-2} p_{i-1}} \cdot \overrightarrow{p_{i-1} p_i} = 1$ and $\overrightarrow{p_{i-1} p_i} \cdot \overrightarrow{p_i p_{i+1}} = 0$ のとき、 (p_{i-1}, p_i, p_{i+1}) を (p_{i-1}, q, p_{i+1}) に置き換える。 q は $(p_{i-1}, q) = (p_i, p_{i+1})$ を満たすルーターとする。(図 10)

一つの更新の中に二つ以上のルーターが含まれていることが分かる。経路 P のどこに局所的な更新が適用されたのかを特定するために、上で述べた更新を p_i での更新と呼ぶこととする。 p_i での更新により経路情報が更新される。 p_i での局所的な更新によって更新しなければならない経路情報を表 3 に示す。

あるルータ p_i において複数の更新を検知することを衝突と呼ぶ。図 11~図 16 にその一覧を示す。衝突を検知した場合、複数の更新を同時に適用すると経路が切断される場合がある。図 17 において、 p_i で shortcut1, zigzag の両方の更新が可能であるが、同時に更新を適用してしまうと図 18 のように経路が切断されてしまう。なので、経路の切断を防ぐために提案アルゴリズムでは局所的な更新の優先順位を導入することによってどちらかの更新が排他的に適用されるようになっている。優先順位は $i < j$ ならば、 p_i が p_j より高い優先順位をもっている。複数の更新が同じルーターで適用可能な場合、shortcut1 と zigzag が同じルーターで適用可能な場合 zigzag を、shortcut2 と zigzag が同じルーターで適用可能な場合 zigzag を優先する。表 4 に衝突した場合の優先順位の一覧を示す。この優先順位を用いて図 16 の経路を更新すると、図 19 のように経路が切断されずに更新される。ただし、図 20 に示すような場合には同時に更新をしても経路は切断されないの優先順位は導入しない。提案アルゴリズムではより高い優先順位の局所的な更新との衝突が存在しない場合にのみ、 p_i における局所的な更新を許可する。

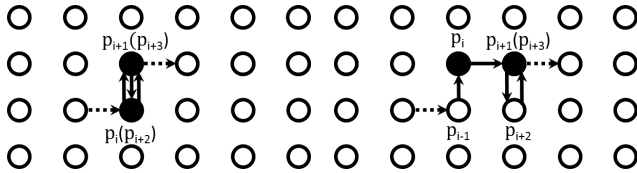


図 11 更新の衝突 1

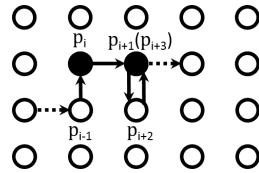


図 12 更新の衝突 2

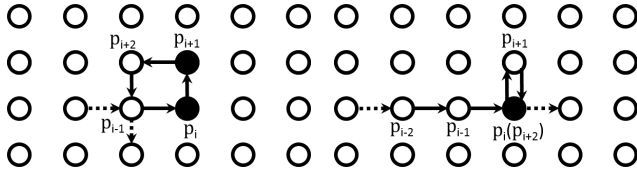


図 13 更新の衝突 3

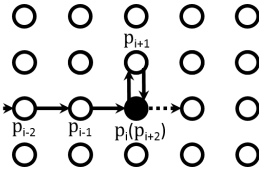


図 14 更新の衝突 4

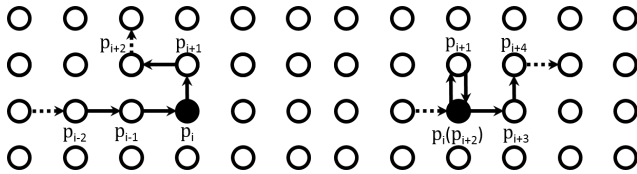


図 15 更新の衝突 5

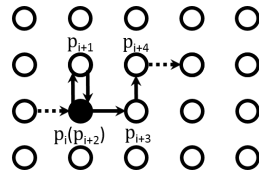


図 16 更新の衝突 6

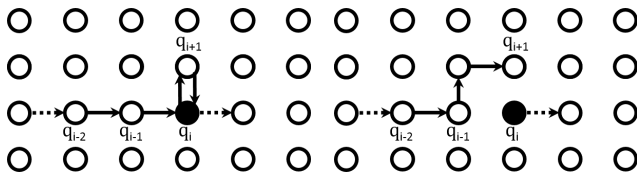


図 17 経路の切断の例 1

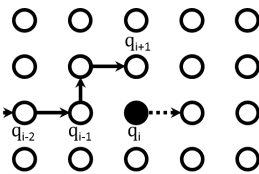


図 18 経路の切断の例 2

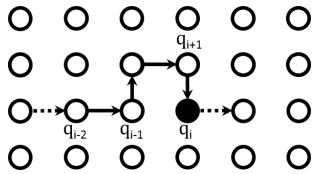


図 19 経路の切断の例 3



図 20 経路が切断されない例

p_i での更新	shortcut1	shortcut2	zigzag
p_{i-1} での優先する更新	shortcut1,2	shortcut2	-
p_i での優先する更新	zigzag	zigzag	-

表 4 衝突した場合の優先順位

2.4 提案アルゴリズムの実行の流れ

提案アルゴリズムでは、経路の情報は経路情報として経路上のルーターに分散して保管されている。提案アルゴリズムは、 $p^s - p^t$ 経路を局所的な更新を繰り返し経路に適用することによって最短の $p^s - p^t$ 経路を形成する。経路情報は提案アルゴリズムの実行の間、経路上のルーターに経路情報として保管されている。局所的な更新が適用可能

Algorithm 1 p_i における提案アルゴリズムの疑似コード

```

var
    state1 ← null:  $p_i$  で shortcut1 が適用可能なかどうかを格納する変数
    state2 ← null:  $p_i$  で shortcut2 が適用可能なかどうかを格納する変数
    state3 ← null:  $p_i$  で zigzag が適用可能なかどうかを格納する変数
function
    1: loop
    2: state1 ← null
    2: state2 ← null
    2: state3 ← null
    3: ルーティング情報を確認する。
    4: if  $\overrightarrow{p_i p_{i+1}} \cdot \overrightarrow{p_{i+1} p_{i+2}} = -1$  then
    5:     state1 ← 1/shortcut1 が適用可能。
    6: else if  $\overrightarrow{p_{i-1} p_i} \cdot \overrightarrow{p_i p_{i+1}} = 0$  and
    7:      $\overrightarrow{p_{i-1} p_i} \cdot \overrightarrow{p_{i+1} p_{i+2}} = -1$  then
    8:     state2 ← 1/shortcut2 が適用可能。
    9: else if  $\overrightarrow{p_{i-1} p_i} \cdot \overrightarrow{p_{i-1} p_i} = -1$  and
    10:     $\overrightarrow{p_{i-2} p_{i-1}} \cdot \overrightarrow{p_i p_{i+1}} = -1$  then
    11:    state3 ← 1/zigzag が適用可能
    12:    局所的な更新の衝突がないかを調べる。
    13:    もし、優先順位の高い更新との衝突が無ければ局所的な更新を適用する。
    
```

かどうかを確認はルーティング情報をもとに行われ、優先順位の高い更新が存在しなければ局所的な更新が適用される。局所的な更新には2つ以上のルーターが使われている。そのため、 p_i は p_i で適用される局所的な更新において、局所的な更新に含まれる他のルーターに対して更新の命令をする。よって、提案アルゴリズムの実行は次のようになる。

- 1) ルーティング情報をもとに、局所的な更新が可能であるかどうかを判断する。
- 2) より優先順位の高い更新がなければ局所的な更新を適用する。このとき、 p_i は関係のあるすべてのルーターに対して更新の命令を送る。

Algorithm1 に p_i における提案アルゴリズムの疑似コードを示す。

2.5 提案アルゴリズムの実行例

提案アルゴリズムの実行の様子を図 21~図 27 に示す。実行の様子は同期して優先順位の高い更新と衝突していない限り局所的な更新が実行されるものを想定している。開始の経路(図 21)の長さは15である。初めの経路では更新 zigzag しか適用できないため、更新によって長さは変わらない(図 22)。2回目の更新では更新 shortcut1 が適用可能であるので長さが減り、長さが11となる(図 23)。6回目の更新で長さが5の最短経路となる(図 27)。適用できる更新がなくなったためアルゴリズムは終了となる。

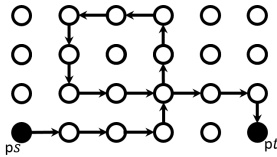


図 21 実行例 1

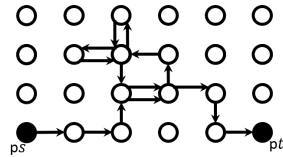


図 22 実行例 2

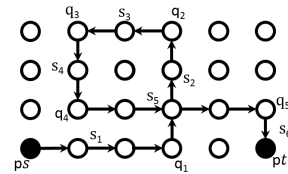


図 28 直線部分

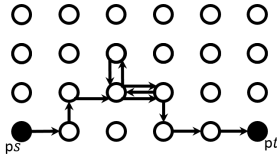


図 23 実行例 3

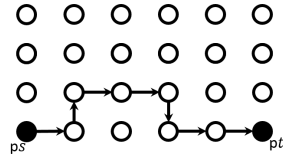


図 24 実行例 4

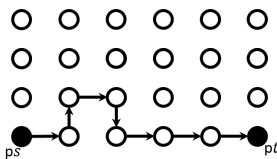


図 25 実行例 5

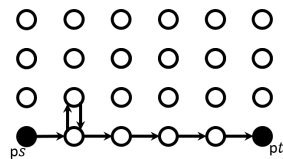


図 26 実行例 6

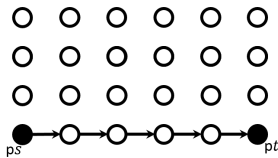


図 27 実行例 7

3. 正当性の証明

提案アルゴリズムが与えられた通信経路を切断することなく最短経路に収束させることを証明するために、提案アルゴリズムが以下の3つのことを満たすことを証明する。

- 1) 提案アルゴリズムの更新が通信経路が切断されない。
- 2) 提案アルゴリズムの更新が経路を最短の通信経路を形成する。
- 3) 更新が経路を最短の通信経路を形成したら、以降は提案アルゴリズムの更新によって経路は変わらない。

補題 1 提案アルゴリズムの実行により通信経路が切断されない

証明 提案アルゴリズムの3種類の更新だけを実行する場合経路が切断されることはない。

経路が切断されるのは複数の更新が同時に行われる場合である。

図 11 から図 16 に更新が衝突する全ての例が挙げられていて、図 16 の更新以外は同時に更新を適用すると経路が切断されてしまう。図 11~図 15 の場合において提案アルゴリズムは次のように経路を更新する。

図 11 の場合には、 p_i と p_{i+1} において shortcut1 を適用することが可能であるが、 p_{i+1} のルータが p_i での shortcut1 を検知することによって p_{i+1} の shortcut1 での更新がされ

なくなる。

図 12 の場合には、 p_i において shortcut2 が、 p_{i+1} において shortcut1 を適用することが可能であるが、 p_{i+1} のルータが p_i での shortcut2 を検知することによって p_{i+1} の shortcut1 での更新がされなくなる。

図 13 の場合には、 p_i において shortcut2 が、 p_{i+1} において shortcut2 を適用することが可能であるが、 p_{i+1} のルータが p_i での shortcut2 を検知することによって p_{i+1} の shortcut2 での更新がされなくなる。

図 14 の場合には、 p_i において shortcut1 と zigzag を適用することが可能であるが、 p_i が shortcut1 と zigzag を検知することによって p_i での shortcut1 の更新がされなくなる。

図 15 の場合には、 p_i において shortcut2 と zigzag を適用することが可能であるが、 p_i が shortcut2 と zigzag を検知することによって p_i での shortcut2 の更新がされなくなる。

全ての場合において経路が切断されないため、提案アルゴリズムの実行によって通信経路が切断されない。□

定義 1 直線部分

$P=(p^s = p_0, p_1, p_2, \dots, p_n = p^t)$ となる経路を $p_s - p_t$ 経路とする。 p^s, p^t または、 $\overrightarrow{p_{i-1}p_i} \cdot \overrightarrow{p_i p_{i+1}} = 0$ または、 $\overrightarrow{p_{i-1}p_i} \cdot \overrightarrow{p_i p_{i+1}} = -1$ となるとき、 p_i を曲ルータ (図 28) と呼ぶ。 p^s, p^t を曲ルータに含めるため、定義より任意の経路 P に 2 個以上存在する。経路 P 中の曲ルータの数を $k+1$ 個とすると、 $q_j (0 \leq j \leq k)$ は経路 P の中で j 番目の曲ルータであり、 $s_j (1 \leq j \leq k)$ を直線部分 (q_{j-1}, q_j) とする。経路 P は直線部分の列 (s_1, s_2, \dots, s_k) で表すことが出来、これを経路 P の直線部分列とする。P を経路とし、 (s_1, s_2, \dots, s_k) を直線部分列とする。 \vec{s}_i はルータのベクトル $\overrightarrow{q_{j-1}q_j}$ と表すことが出来る。

定義 2 ポテンシャル関数の定義

shortcut1 を適用するルータが存在する場合、 b を $\vec{s}_i \cdot \overrightarrow{s_{i+1}} = -1$ を満たす最小の i とする。 ($1 \leq i \leq k$) において、 $i=b$ または $i=b+1$ のとき、 ($0 < \alpha < 1$) となる α を用いて $|s_i'| = |s_i| - 1 + \alpha$ とする。 $i=b$ または $i=b+1$ 以外のとき、 $|s_i'| = |s_i|$ とする。 経路 P を直線部分列が (s_1, s_2, \dots, s_k) の経路とする。ポテンシャル関数 f は $f(P) = (h, |s_1'|, |s_2'|, \dots, |s_k'|)$ (h は経路 P の全体の長さであり、 $|s_i| (1 \leq i \leq k)$ を直線部分 s_i の長さとする。

shortcut1 を適用するルータが存在しない場合、ポテン

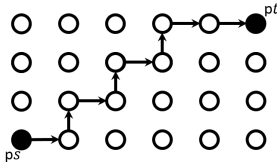


図 29 最短経路 1

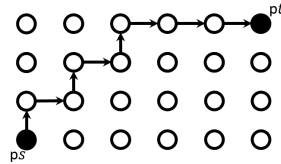


図 30 最短経路 2

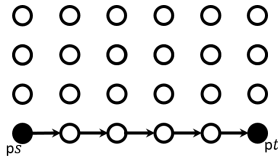


図 31 最短経路 3

シャル関数 f は $f(P) = (h, |s_1|, |s_2|, \dots, |s_k|)$ (h は経路 P の全体の長さであり, $|s_i| (1 \leq i \leq k)$ を直線部分 s_i の長さとする.

($h = \sum_{i=1}^k |s_i|$ が成り立つ.) ポテンシャル関数は辞書順でソートされるものとする.

定義 3 最短経路の定義

P が最短経路であるならば, 直線部分列 (s_1, s_2, \dots, s_k) は $(0 \leq i \leq k-1)$ において $|s_i| = 1$ を満たす. 通常, 1つの直線部分で構成されていなければ, 最短経路は図 29, 図 30 に示すように 2つ存在する. 一つの直線部分で構成されている場合, 最短経路は図 31 に示すようにただ一つ存在する.

補題 2 経路の更新によって経路のポテンシャル関数が減少する.

補題 2 の証明

一番先頭で適用されるの更新が, shortcut1 の場合, 長さ自体が短くなるため, ポテンシャル関数の初めの要素が減少する. よって, 更新によりポテンシャル関数の値が減少する.

一番先頭で適用されるの更新が, shortcut2 の場合, $P = (p^s = p_0, p_1, p_2, \dots, p_n = p^t)$ とし, p_a を一番先頭の shortcut2 が適用されるルーターの引数とする. (s_1, s_2, \dots, s_k) を P の直線部分列とし, s_b を p_a を含む直線部分とする. $P' = (p^s = p'_0, p'_1, p'_2, \dots, p'_n = p^t)$ とし, $(s'_1, s'_2, \dots, s'_k)$ を P' の直線部分列とする.

$(0 \leq i \leq b-2)$ のとき, $|s'_i| = |s_i|'$

$(i=b-1)$ のとき, $|s'_i| = |s_i|' - 1 + \alpha$ となる. よって, 更新によりポテンシャル関数の値が減少する.

一番先頭で適用されるの更新が, zigzag の場合, $P = (p^s = p_0, p_1, p_2, \dots, p_n = p^t)$ とし, p_a を一番先頭の zigzag が適用されるルーターの引数とする. (s_1, s_2, \dots, s_k) を P の直線部分列とし, s_b を p_a を含む直線部分とする. $P' = (p^s = p'_0, p'_1, p'_2, \dots, p'_n = p^t)$ とし, $(s'_1, s'_2, \dots, s'_k)$ を P' の直線部分列とする.

$(0 \leq i \leq b-1)$ のとき, $|s'_i| = |s_i|'$

$(i=b)$ のとき, $|s'_i| = |s_i|' - 1$ となる. よって, 更新によりポテンシャル関数の値が減少する. □

補題 3 最短経路のポテンシャル関数が最小 (あるいは 2 番目に小さい値) になる.

補題 3 の証明 定義 2, 定義 3 より, 最短経路のポテンシャル関数は最小になる. □

補題 4 どんな通信経路でも最短経路を形成する

補題 4 の証明 補題 1 は提案アルゴリズムの実行の間経路が切断されないことを示している. 補題 3 はポテンシャル関数が単調減少になることを示している. 補題 2 は最短の $p^s - p^t$ 経路は他の $p^s - p^t$ 経路よりも小さいポテンシャル関数を保持していることを示している. よって, 初めに与えられた経路よりも小さいポテンシャル関数の $p^s - p^t$ 経路の数は有限であり, このことは経路が最短経路でない限り経路に局所的な更新を適用することが可能であることを示している. P を最短でない $p^s - p^t$ 経路とし, その直線部分列を (s_1, s_2, \dots, s_k) とする.

- $s_i (1 \leq i \leq k-1)$ において, $|s_i| \geq 2$ を満たす s_i が存在するならば, 更新 zigzag を適用することが可能である.

- $s_i (1 \leq i \leq k-1)$ の全ての i において, $|s_i| \geq 1$ を満たすならば, P が最短経路でないので次の 2つの場合が考えられる.

- 1) $(1 \leq i \leq k-1)$ において $\vec{s}_i \cdot \vec{s}_{i+1} = -1$ を満たす i が存在するならば, 更新 shortcut1 を適用することが可能である.

- 2) $(1 \leq i \leq k-1)$ の全ての i において $\vec{s}_i \cdot \vec{s}_{i+1} = 0$ を満たすならば, 更新 shortcut2 を適用することが可能である.

このように, 経路 P に対して更新を適用することが可能である. 適用される更新はより高い優先順位の更新と衝突しない更新である. □

補題 5 最短経路が形成されたら更新が行われない

補題 5 の証明 更新 shortcut1 または更新 shortcut2 を適用するならば, $\vec{s}_i \cdot \vec{s}_j$ を満たす s_i と s_j が存在する. しかし, 補題より P が最短経路のとき, そのような直線部分は存在しない. 更新 zigzag が適用可能のとき, $(1 \leq i \leq k-1)$ において $|s_i| \geq 2$ を満たす s_i が存在する. しかし, 補題より P が最短経路の時そのような直線部分は存在しない. よって, 補題が成り立つ. □

定理 1 提案アルゴリズムは通信経路を切断することなく最短経路に収束させる

補題 1 は提案アルゴリズムの更新が通信経路が切断されないことを証明している. 補題 4 は提案アルゴリズムの更新が通信経路が最短経路を形成することを証明している. 補題 5 は通信経路が最短経路に形成したら以降は提案アルゴリズムの更新によって経路は変わらないことを証明している. よって, 提案アルゴリズムは通信経路を切断することなく最短経路に収束させる.

4. 評価実験

Zigzag プロトコルと提案アルゴリズムの収束時間の差を

経路長	100	1000	10000
zigzag プロトコル	13.261	48.379	234.499
提案アルゴリズム	17.097	57.241	264.896

表 5 評価実験の結果 1

経路長	100	1000	10000
収束時間の变化の割合 (%)	128.93	118.32	112.96

表 6 評価実験の結果 2

検証するため、評価実験を行う。Zigzag プロトコルは計算時間 $O(|P|)$ ($|P|$ は経路の初期状態の長さ) で経路を最短経路に収束させることが出来る。今回の実験では、グリッドのサイズは無限であると仮定し、経路の生成はグリッドネットワーク上の自分がある地点の上下左右のルータの中からランダムに選択しそのルーターに経路を伸ばすことを繰り返し生成するものとする。優先順位を考慮して更新できる更新をすべて適用し新たな経路を構成することを 1 ラウンドと呼ぶものとし、各アルゴリズムで最短経路に収束するまでのラウンド数を測定し、平均ラウンド数を得る。1000 回シミュレーションした結果を表 5 に示す。

実験の結果より、提案アルゴリズムは zigzag プロトコルよりも収束時間が長くなるということが得られた。また、表 6 に示すように経路長が長くなるほど zigzag プロトコルの収束時間に近づいていくことが得られた。

5. おわりに

本論文では仮想グリッド上での経路最適化問題を解決する既存研究の Zigzag アルゴリズムで解決する問題を経路上 2 ホップに限定して解決するアルゴリズムを提案した。提案アルゴリズムでは、視野を 2 ホップに限定することによって、視野の範囲の情報を保存するメモリを少なくすることが出来たり、視野の範囲の情報を取得する際の通信数を少なくすることが出来るというメリットがある。今後の課題には提案アルゴリズムの計算時間の解析、別のアルゴリズムの条件では、視野が経路上 1 ホップに限定されたものやターゲットが 1 つではなく複数あるものなどが挙げられる。

6. 参考文献

参考文献

- [1] Walteneus Dargie and Christian Poellabauer. Fundamentals of Wireless Sensor Networks: Theory and Practice, page 187. John Wiley & Sons, Ltd, 2010.
- [2] Carlos de Morais Cordeiro & Dharma Prakash Agrawal. Ad Hoc and Sensor Networks Theory and Applications (2nd Edition). World Scientific, 2011.
- [3] Silvia Giordano. Mobile Ad-Hoc Networks. Wiley [Imprint], Inc., 2000.
- [4] Brad Karp and H. T. Kung. Gpsr: Greedy perimeter-stateless routing for wireless networks. In AC Mobicom, 2000.
- [5] C.E. Perkins and E.M. Royer. Ad hoc on demand dis-

- tance vector routing. In Workshop Mobile Computing Systems and Applications (WMCSA '99), 1999.
- [6] Ya Xu, John Heidemann, and Deborah Estrin. Geography-informed energy conservation for ad hoc routing. In Proceedings of the Seventh Annual ACM/IEEE International Conference on Mobile Computing and Networking (ACM Mobicom), 2001.
- [7] C. de Morais Cordeiro and D. P. Agrawal, Ad Hoc and Sensor Networks: Theory and Applications (2nd Edition). World Scientific, 2011.
- [8] S. Giordano, Mobile Ad-Hoc Networks. Wiley, 2000.
- [9] J. Zheng and A. Jamalipour, Wireless Sensor Networks: A Networking Perspective. Wiley-IEEE Press, 2009.
- [10] Y. Xiao, H. Chen, and F. H. Li, Handbook on Sensor Networks. World Scientific, 2010.
- [11] W. Dargie and C. Poellabauer, Fundamentals of Wireless Sensor Networks: Theory and Practice. John Wiley & Sons, Ltd, 2010.
- [12] B. Karp and H. T. Kung, "GPSR: Greedy Perimeter Stateless Routing for wireless networks," in Proceedings of the 6th Annual International Conference on Mobile Computing and Networking (MobiCom '00), 2000, pp. 243-254.
- [13] C. Perkins and E. Royer, "Ad hoc on demand distance vector routing," in Proceedings of the 2nd IEEE Workshop on Mobile Computing Systems and Applications (WMCSA '99), 1999, pp. 90-100.
- [14] Y. Xu, J. Heidemann, and D. Estrin, "Geography-informed energy conservation for ad hoc routing," in Proceedings of the 7th Annual International Conference on Mobile Computing and Networking (MobiCom '01), 2001, pp. 7084.
- [15] C. E. Perkins and P. Bhagwat, "Highly dynamic Destination- Sequenced Distance-Vector routing (DSDV) for mobile computers," in Proceedings of Conference on Communications Architectures, Protocols and Applications (SIGCOMM '94), 1994, pp. 234-244.
- [16] D. Braginsky and D. Estrin, "Rumor routing algorithm for sensor networks," in Proceedings of the 1st ACM International Workshop on Wireless Sensor Networks and Applications (WSNA '02), 2002, pp. 22-31.
- [17] T. H. Clausen, G. Hansen, L. Christensen, and G. Behrmann, "The optimized link state routing protocol evaluation through experiments and simulation," in Proceedings of IEEE Symposium on Wireless Personal Mobile Communications, 2001.
- [18] S. Takatsu, F. Ooshita, H. Kaskugawa, T. Masuzawa, Zigzag: Local-Information-Based Self-Optimizing Routing in Virtual Grid Networks, IEEE 33rd International Conference on Distributed Computing Systems (ICDCS), pp. 357-368, 2013.