

均一で密なユニットディスクグラフにおける 局所情報に基づく経路の自己構成手法

東優樹 大下福仁 角川裕次 増澤利光
大阪大学 大学院情報科学研究科

概要 本稿では、均一で密なユニットディスクグラフにおいて、任意のホップ数冗長経路から最短経路の高々定数倍の経路を構築する手法について考察する。均一で密とは、ノードの通信可能範囲を8等分の領域に分割したとき各領域に少なくとも1つのノードが存在することを意味する。送信ノードから宛先ノードへの経路において宛先ノードから遠ざかる移動がなければ冗長な経路が少ないことから、8等分された領域のうち連続した2つの領域のみで経路を構築することを考える。各ノードが局所情報のみを利用し、自律的に経路を変更することによって、目的の経路を構築する手法を紹介する。また、提案アルゴリズムによって得られた経路の最短経路に対する近似率について考察する。

1 はじめに

近年、無線デバイス（ノード）の普及により、分散システムにおいて無線ネットワークは一般的かつ重要となっており、例としてMANET（モバイルアドホックネットワーク）やWSN（ワイヤレスセンサネットワーク）がある。無線ネットワークでは、ノードは二次元平面に配置されており、通信範囲内のノードとのみ直接通信可能である。宛先ノードが通信範囲外のときは、他のノードが宛先ノードへのメッセージを中継する。通信経路が長いほどメッセージ送信に伴う通信遅延や消費エネルギーが増加する。効率的なマルチホップルーティングを実現するために、数々のルーティングプロトコルが提案されている [1]。

送信ノードから宛先ノードへの最短経路を構築する手法として、仮想グリッドネットワーク上でルーティングを行う手法が提案されている [2]。しかし、仮想グリッドネットワークの構築やノードの移動時の処理にはコストがかかる。本稿では、ユニットディスクグラフ上で任意のホップ数冗長経路から最短経路の高々定数倍の経路を構築するアルゴリズムについて考察する。均一で密とは、ノードの通信可能範囲を8等分の領域に分割したとき各領域に少なくとも1つのノードが存在することを意味する。

経路の冗長な部分を減らすには、送信ノードから宛先ノードへの経路において宛先ノードから遠ざかる方向への経路が存在しなければ良い。そこで、送信ノードから宛先ノードへ8分割された領域のうち連続した2つの領域のみで到達する経路を構築する手法を考える。本稿では、各ノードが局所情報に基づいて、経路切断が発生しないように経路を局所的に変更するという動作を繰り返すことにより、与えられた通信経路を最短経路の高々定数倍の経路に変換するアルゴリズムの設計への取り組みを紹介する。

2 諸定義

本稿では、ユニットディスクグラフ $G = (V, E)$ で

の送信ノード v_s から宛先ノード v_t へのメッセージ転送を考える。ここで、 V はノードの集合、 E は隣接ノード間を結ぶ通信リンクの集合を表す。ネットワークのノード数を $n = |V|$ とし、各ノードの識別子を $v_i (0 \leq i \leq n-1)$ とする。また、ノード v_i, v_j 間の通信リンク $(v_i, v_j) \in E$ は $v_i, v_j \in V$ に対し、 $|v_i - v_j| < 1$ が成り立つとき、かつそのときのみ存在する。ただし、 $|v_i - v_j|$ は v_i と v_j のユークリッド距離を表す。また、ユニットディスクグラフ G に対し、以下の2つの制限を加える。

1. 各ノードは4ホップ先までのノードの情報を知っている。
2. 各ノードは距離 $\frac{1}{\sqrt{2}}$ 以下に8方向の辺を持つ (図1)。つまり、 0.7 の各領域に対して少なくとも1つのノードが存在する。また、全ノードが同じ方向感覚を有するものとする。

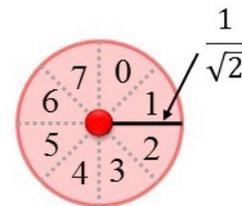


図1: ノードの方向感覚

送信ノード v_s から宛先ノード v_t までの通信経路を $P = (v_s = p_0, p_1, \dots, p_m = v_t)$ と表し、辺 (p_i, p_{i+1}) の方向 d_i を p_i から見た p_{i+1} の存在する領域とする。

定義 1 (通信経路準最適化問題). v_s, v_t をユニットディスクグラフ G の異なるノードとしたとき、任意の $v_s - v_t$ 経路が与えられたら、最短 $v_s - v_t$ 経路の高々 α 倍以下の $v_s - v_t$ 経路を構築する。このとき、 α は最短 $v_s - v_t$ 経路に対する近似率である。□

$v_s - v_t$ 経路 P 上のノード v_i が所持する情報を以下に示す。 v_i が P 上に k 回出現するとき、 v_i は k 個の組 $(pre_1, next_1), (pre_2, next_2), \dots, (pre_k, next_k)$ を所持する。 P 上の p_i を v_i の j 番目の出現とすると、 v_i が保持する j 番目の組 $(pre_j, next_j)$ は (p_{i-1}, p_{i+1}) である。

3 提案アルゴリズム

提案アルゴリズムは3つの操作(操作1, 操作2, 操作3)から構成され, これらの操作を繰り返すことで任意の経路を準最適経路へ変換する.

操作1 局所的に冗長な部分を削除

操作2 局所的に連続した3方向以上の経路を連続した2方向以下の経路に変換

操作3 局所的に連続した3方向以上の経路を検出できるように経路を変換

提案アルゴリズムは以下のように動作する.

1. 経路上の各ノード p_i は周囲の経路の情報を取得し, p_i において局所変更が可能かどうかを調べる.
2. p_i において局所変更が可能であれば適用し, 1.に戻る. そうでなければそのまま1.の処理に戻る. ただし, 経路切断の発生を回避するため, 局所変更適用時は近隣ノードと強調して, 実際に適用するかどうか決定する.

Algorithm 1 提案アルゴリズム:for ノード p_i

- 1: **loop**
- 2: 近隣1ホップにあるノードの所持する経路情報を取得し, p_i において局所変更が可能か調べる. ;
- 3: **if** p_i において局所変更が可能 **then**
- 4: 近隣ノードが局所変更を実行可能か調べる ;
- 5: **if** 局所変更を実行しても経路切断が発生しない **then**
- 6: 実行可能な局所変更を実行 ;
- 7: **goto** 2 ;
- 8: **else**
- 9: **goto** 2 ;

3.1 操作1

操作1では, 局所的にホップ数が冗長な経路に対して冗長部分を削除する変更を行う. 操作1の局所変更を以下に示す.

定義2 (冗長部分の除去).

update1: 辺の往復の削除 (図2(a))

辺 $(p_{i-1}, p_i), (p_i, p_{i+1})$ について $p_{i-1} = p_{i+1}$ ならば, $(p_{i-1}, p_i), (p_i, p_{i+1})$ を削除する. ただし, $d_{i-2}, d_{i-1}, d_i, d_i + 1$ が連続した2方向のみから成る経路の場合は適用しない.

update2: ホップ数の削減 (図2(b))

辺 $(p_{i-1}, p_i), (p_i, p_{i+1})$ について $(p_{i-1}, p_{i+1}) \in E$ ならば, $(p_{i-1}, p_i), (p_i, p_{i+1})$ を削除し, (p_{i-1}, p_{i+1})

を追加. また, 辺 $(p_{i-1}, p_i), (p_i, p_{i+1}), (p_{i+1}, p_{i+2})$ について $(p_{i-1}, p_{i+2}) \in E$ ならば, $(p_{i-1}, p_i), (p_i, p_{i+1}), (p_{i+1}, p_{i+2})$ を削除し, (p_{i-1}, p_{i+2}) を追加する.

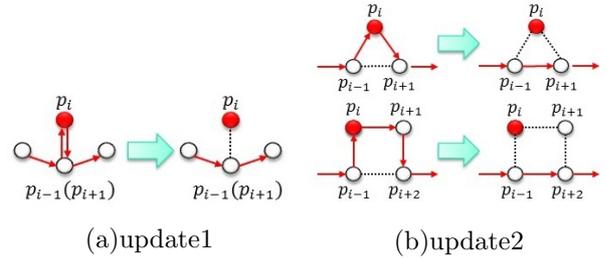


図2: 冗長部分の削除

3.2 操作2

操作2では, 局所的に連続した3方向以上の経路を連続した2方向以下の経路に変更する. 各 update では, 変更後の経路は連続した2方向以下の経路を選択するものとする. 操作2の局所変更の適用条件を以下に示す.

定義3 (方向の削減).

update3: 1つ飛ばし (図3(c))

辺 $(p_{i-1}, p_i), (p_i, p_{i+1})$ について, d_{i-1} と d_i が1つ飛ばし(連続する3方向の経路). つまり, $|d_{i-1} - d_i| = 2$ もしくは $|d_{i-1} - d_i| = 6$.

update4: 2つ飛ばし (図3(d))

辺 $(p_{i-1}, p_i), (p_i, p_{i+1})$ について, d_{i-1} と d_i が2つ飛ばし(連続する4方向の経路). つまり, $|d_{i-1} - d_i| = 3$ もしくは $|d_{i-1} - d_i| = 5$.

update5: 折り返しが存在 (図3(e))

辺 $(p_{i-1}, p_i), (p_i, p_{i+1}), (p_{i+1}, p_{i+2})$ について, d_{i-1}, d_i, d_{i+1} は連続しているが d_{i-1} と d_{i+1} が反対方向に進んでいる. つまり, $|d_{i-1} - d_i| \% 6 = 1$ かつ $|d_i - d_{i+1}| \% 6 = 1$ かつ $|d_i + 1 - d_{i+2}| \% 4 = 2$.

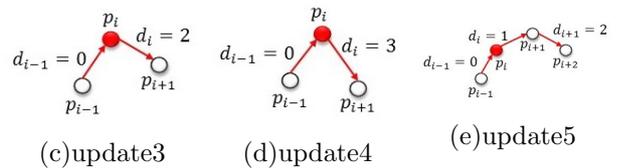


図3: 3方向以上の経路を削除

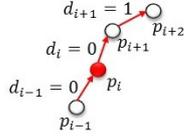
3.3 操作3

操作3では, 局所的に連続した3方向以上の経路を検出できるように変更する. 操作3の局所変更を以下に示す.

定義 4 (3 方向以上の経路の検出).

update6: 同じ方向の後に隣接方向に移動 (図 4 (f))

辺 $(p_{i-1}, p_i), (p_i, p_{i+1}), (p_{i+1}, p_{i+2})$ について, d_{i-1} , d_i の方向が同じかつ d_i, d_{i+1} が隣接した方向である. つまり, $d_{i-1} = d_i$ かつ $|d_i - d_{i+1}| \% 6 = 1$.



(f)update6

図 4: 3 方向の検出

連続した 3 方向以上の経路の検出

update6 を繰り返し実行することで局所的に連続した 3 方向以上の経路を検出することができる (図 5).

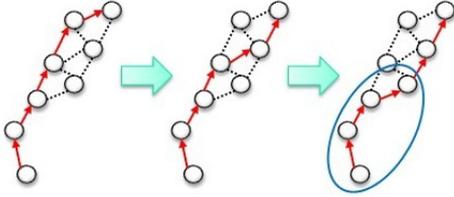


図 5: 局所的に 3 方向を検出する様子

3.4 局所変更の優先度

経路 P 上のノード p において複数の局所変更が実行可能な場合が起こりうる. その場合は以下の優先度に従い優先度の高い局所変更を実行する.

定義 5 (局所変更の優先度). $update1 > update2 > update3 > update4 > update6 > update5$ □

また, 連続したノードで局所変更が実行されると経路が非連結になる可能性があるため経路の前後関係での局所変更の優先度を次のように定める, ノード p_i は前後 2 ホップのノード $(p_{i-2}, p_{i-1}, p_{i+1}, p_{i+2})$ で自身より優先度の高い変更が実行可能ならば局所変更を実行しない. また前 2 ホップのノード (p_{i-2}, p_{i-1}) で自身と同じ優先度の変更が実行可能ならば局所変更を実行しない.

4 近似率

提案アルゴリズムにより最終的に得られる連続した 2 方向のみの準最短経路 P の近似率について考察する. 経路を連続した 2 方向のみの経路に変更することで, 図 6 のように経路の存在する範囲を限定することができる. ここで, $|v_s - v_t| = r$, v_s と v_t のなす角を θ と

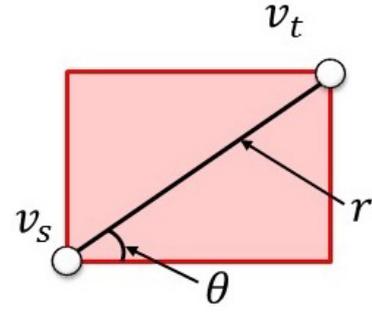


図 6: 経路の存在範囲

する. このとき, 経路 P が宛先ノード v_t から遠ざかる方向に移動しないこと, 少なくとも 2 ホップで距離 1 以上 v_t に近づくことを考慮するとホップ数と距離による近似率は表 1 のようになる.

表 1: 近似率

| | 最短 | 最長 | 近似率 |
|------|-----|---|-------------|
| ホップ数 | r | $2\sqrt{2}r \sin(\theta + \frac{\pi}{4})$ | $2\sqrt{2}$ |
| 距離 | r | $\sqrt{2}r \sin(\theta + \frac{\pi}{4})$ | $\sqrt{2}$ |

5 まとめと今後の課題

本稿では, 均一で密なユニットディスクグラフにおいて, 送信ノードから宛先ノードへの通信経路が与えられたとき, 各ノードが局所的な情報のみを用いて局所的に通信経路を変更することによって, 最短経路の高々定数倍となる経路を構築する手法を提案した. 提案アルゴリズムによって得られる経路の最短経路に対する近似率はホップ数で $2\sqrt{2}$, 距離で $\sqrt{2}$ となる.

今後の課題としては, 収束時間の評価と 8 方向のうち辺が存在しない方向が存在する場合に対するアルゴリズムの考案が挙げられる.

参考文献

- [1] W. Dargie and C. Poellabauer. *Fundamentals of Wireless Sensor Networks: Theory and Practice*. Wireless Communications and Mobile Computing. Wiley, 2010.
- [2] Shusuke Takatsu, Fukuhito Ooshita, Hirotsugu Kakugawa, and Toshimitsu Masuzawa. *Zigzag: Local-Information-Based Self-Optimizing Routing in Virtual Grid Networks*. In ICDCS, pp. 357–368, 2013.