

動的な環境下における履歴情報を用いた確率的ルーティング

齊藤 亨[†] 松尾 啓志^{††}

[†] 名古屋工業大学電気情報工学科
^{††} 〒466-8555 愛知県名古屋市昭和区御器所

E-mail: [†]asaito@mars.elcom.nitech.ac.jp, ^{††}matsuo@elcom.nitech.ac.jp

あらまし 近年、携帯電話や PDA 等、小型で高性能な携帯端末が開発される中で移動体通信が生活の中で欠かせないものとなりつつある。そのような中で注目されているのがアドホックネットワークである。アドホックネットワークのような動的にトポロジが変化する環境では、固定的なネットワークにおいて用いられてきた最適な経路を維持しようとする手法では柔軟なルーティングを行うことが難しい。そこで、確率統計的に良好な経路を選択する手法が提案されている。しかし、それらの手法は経路を学習する速度が遅いという欠点があった。本研究ではアドホックネットワークにおいて確率的な手法を用い、素早く経路を学習することができるルーティングアルゴリズムの提案と評価を行う。

キーワード 動的環境, アドホックネットワーク, 強化学習

An Adaptive Ant-based Routing Algorithm used routing history in Dynamic Networks

Akira SAITO[†] and Hiroshi MATSUO^{††}

[†] Department of Electrical and computer Engineering, Nagoya Institute of Technology, Gokiso, Showa-ku, Nagoya, 466-8555, Japan

E-mail: [†]asaito@mars.elcom.nitech.ac.jp, ^{††}matsuo@elcom.nitech.ac.jp

Abstract In recent year, we can use the mobile communication in our life according to development of high performance cellular phone, personal data assistant (PDA) and so on. In such a scene, Ad-hoc network has been spotlighted. However conventional routing algorithm that effort to keep optimal paths is difficult to apply to dynamic topology network such as ad-hoc network. Therefore the techniques that select good routing paths with probability rule are proposed. But those convergence speed and routing result are still unsatisfied. In this paper, we propose new routing algorithm that uses routing history, and obtain good routing path with probability rule. Fairly good convergence speed and good routing path is achieved.

Key words dynamic networks, ad-hoc network, reinforcement learning

1. アドホックネットワーク

近年、携帯電話が爆発的に普及し、電子メールや Web アクセス等、移動体通信において提供されるサービスもますます多様化している。そのような中で、新たなコミュニケーション環境の媒体として、アドホックネットワークが注目を集めている。アドホックネットワークは必要に応じて一時的に構築するネットワークであり、いつでも、どこでも自由に移動端末(以下、ノードと呼ぶ)間で情報の交換・共有を行うことを可能とする。

本研究で対象としているアドホックネットワークの特徴を以下に示す。

- (1) 新たなノードの参加や退出など、ネットワークトポロジの動的な変化がある
- (2) 無線マルチホップ通信により通信を行う
- (3) ネットワークを集中管理する端末が無い
- (4) 各ノードは同等の機能を持ち、自律的に双方向通信可能なネットワークを構築する

本研究では、このような環境の中で1対1の通信を行う際、効率的に経路情報を獲得し、適した経路を選択するルーティングアルゴリズムを提案する。

2. 関連研究

2.1 IETFの各手法

アドホックネットワークにおけるルーティングアルゴリズムは The Internet Engineering Task Force (IETF) の Mobile Ad-hoc Network (MANET) Working Group [1] により多数提案されている。提案されている手法は主に次の2種類に大別できる。

1つは定期的なパケット交換によりネットワークトポロジをテーブルに記録し、通信時にその情報に従って最適なルーティングを行う方法である。経路情報の維持に必要な制御パケットの通信量が多く、トポロジの変化に即応することが難しいため、ノードの移動速度が遅く、パケットの発生量が多い場合には有効な手法である。

もう1つは、通信が発生した際にフラッディングを用いて経路情報を獲得する方法である。通信時に大量の無駄なパケットが発生するという欠点があるため、前者とは逆にノードの移動速度が速く、パケットの発生量が少ない場合には有効な手法である。

2.2 強化学習の手法

IETFの手法とは異なるアプローチで動的な環境に対応する手法として、Q-routing [2] や DRQ-routing [3] 等の、強化学習をルーティングに応用した手法が

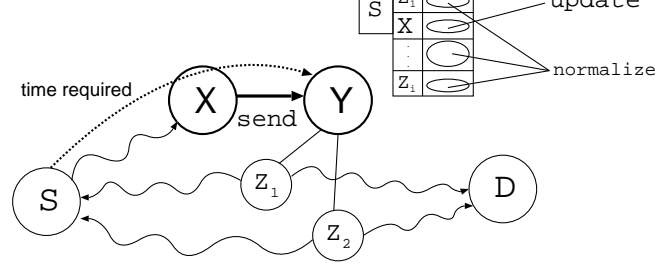


図1 Ants routingの学習

提案されている。その中でも、確率的に良好な経路を選択する Ants routing [4] や Accelerated Ants routing [5] は、最適な経路を常に保持しようとする手法に比べ、トポロジの変化に対する適応力に優れている。

Ants routing は、各ノードが、パケットの宛先に対し次に送信すべきノード(以下、Next hop と呼ぶ)を決定するための選択確率をルーティングテーブルに保持する。

学習は、backward exploration と呼ばれる手法で行う。これはパケットが送信されてきた経路を逆に進む経路について学習するものである。図1に学習の例を示す。ノード x からノード y にメッセージが送信されたとき、 y のテーブルが更新される。更新される確率値を P_y とすると、以下のような学習式で P_y は更新される。

$$\Delta p = k/f(c) \tag{1}$$

$$P_y(s, x) = \frac{P_y(s, x) + \Delta p}{1 + \Delta p} \tag{2}$$

$$P_y(s, z) = \frac{P_y(s, z)}{1 + \Delta p} \quad z \in \text{neighbor of } y, z \neq x \tag{3}$$

ここで、 s はメッセージの送信元ノード、 k は学習率、 $f()$ は単調非減少関数、 c はメッセージが作成されてからの経過時間である。 s を宛先としたパケットの Next hop に x を選択する確率を更新し、その他のノード z_i の選択確率を正規化する。

Accelerated Ants routing は Ants routing に「No return rule」と「N step backward exploration」の2つの機能を付加することにより学習効率を高めたものである。

No return rule とは、Next hop を決定する際、そのパケットを自ノードへ送信したノードは Next hop の選択候補から除外するものである。また、N step backward exploration は、経由した N 個のノードの履歴をメッセージパケットに格納しながら送信することで、送信元ノードだけでなく、中継したノードを宛先とした学習も同時に行うものである。

3. 従来手法の問題点

強化学習的手法の最も大きな問題点は、学習速度が遅く、初期の学習に非常に多くのパケットが必要であることである。特に、ノードの数が多い場合や、ノードの移動が激しい場合には学習が追いつかず、適した経路に収束しないこともあった。

また、Ants routing や AntNet 等、確率的ルーティング特有の問題に routing-lock がある [6]. routing-lock とは、学習が進むことによりルーティング確率が 1 付近に収束し、経路が固定化されてしまう現象である。この現象により、次の 2 つの問題が発生する。

1 つは、routing-lock が発生した後、トポロジの変化が起こり、その経路が利用できなくなった場合に新たな経路を発見するまでに非常に多くの時間が必要となる問題である。これをブロッキング問題と呼ぶ。

もう 1 つは、同じく routing-lock が発生した後、トポロジの変化が起こり、新たに効率の良い経路が生成された場合でも、その経路を発見することができず、従来の経路を選択しつづけてしまうという問題である。これをショートカット問題と呼ぶ。

そこで本論文では Ants routing と AntNet [7] をベースとして、メッセージパケットに履歴情報を記録することでより学習が速く、適応力の高いルーティングアルゴリズムを提案する。

4. 提案手法

4.1 経路選択

Ants routing と同様、各ノードはメッセージパケットを送信する際、テーブルを参照し、その確率値に従って Next hop を決定する。確率値は次の正規化条件を満足している。

$$\sum_{n \in \text{neighbor of } k} P_k(d, n) = 1, d \in [1, N] \quad (4)$$

ここで、 $P_k(d, n)$ は宛先ノード d のとき、ノード k がノード n を Next hop として選択する確率である。

4.2 更新手順

各ノードは、メッセージパケットを中継する際に、自ノードの ID と、伝送遅延やメッセージ処理等の所要時間の合計を履歴としてメッセージパケットに格納し、Next hop に送信する (図 2)。メッセージ内における履歴を格納するキューのサイズは一定であり、古いものから順に削除される。これは昔に経由したノードであればあるほど、その経路が変化している可能性が高くなるためである。これにより、新しく、

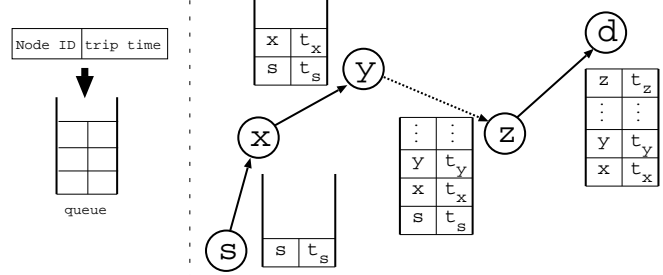


図 2 履歴情報の伝送

信頼性の高い情報のみを残すことができる。

メッセージパケットを受信したノードは backward exploration により学習を行う。学習は、メッセージパケットに格納されている全ての中継ノードを宛先としたルーティングテーブルに対して行う。ノード y が宛先 d のパケットの Next hop にノード n を選択する確率を $P_y(d, n)$ とすると、ノード x からノード y にメッセージが送信された際の学習は次式で表される。

$$\Delta p = \gamma^h \cdot \frac{k}{f(t_{d'})} \quad \gamma \in (0, 1), k > 0 \quad (5)$$

$$P_y(d', x) = \frac{P_y(d', x) + \Delta p}{1 + \Delta p} \quad (6)$$

$$P_y(d', z) = \frac{P_y(d', z)}{1 + \Delta p} \quad z \in \text{neighbor of } y, z \neq x \quad (7)$$

ここで、 k は学習率、 $f()$ は単調非減少関数、 d' は中継ノード、 $t_{d'}$ は d' から x までの所要時間、 γ は割引率、 h は y から d' までのホップ数である。

式 (5) の Δp は確率値の変化量を表す。これにより、式 (6) において送信されてきたノードを Next hop に選択する確率を更新し、式 (7) においてその他のノードの正規化を行う。確率値の変化量 Δp は一定値 k に所要時間 $t_{d'}$ の逆数および割引率 γ を掛けることによって、より近く、信頼性の高い経路に大きい変化量を割り当てる。これは強化学習における報酬の伝播にヒントを得たものであり、これにより、適切な変化量を与え、効率的な学習を行うことを可能とする。

4.3 上限値付き提案手法

ここまで述べてきた提案手法は、Ants routing や AntNet と同じく、3 章で述べた routing-lock が問題となる。そこで「確率値に一定の上限を設け、それ以上の学習をキャンセルする」ルールを追加する。

具体的には、式 (6) で確率値を計算した結果、確率の上限値 $limit$ を超えることが予測された場合、確

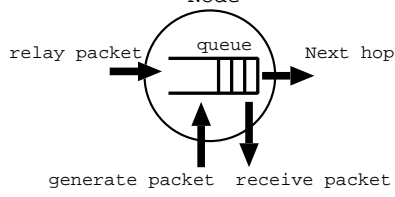


図3 ノードモデル

率値の変化量 Δp を次の式に置き換える。

$$\Delta p = \frac{limit - p}{1 - limit} \quad (8)$$

これにより、学習後の確率値が上限値 $limit$ になるように抑制する。

本稿ではこれ以降、このルールを追加していないものを「提案手法」、追加したものを「上限値付き提案手法」と呼ぶことにする。

5. 実験と評価

提案手法、及び上限値付き提案手法の性能を評価するためにシミュレーション実験を行った。実験は、OSPF, AntNet, Ants routing, Accelerated Ants routing(以下, Acc.Ant と略す)との比較を行った。

5.1 ノードモデル

比較対象とするそれぞれの手法はそれぞれ異なる環境で評価を行っているため、実験環境が厳密には異なる。そこで実験環境を統一するために以下のようなノードモデルで実験と評価を行う。

まず、ノードを単純にモデル化し、1単位時間(以下、シミュレーションステップまたはステップと呼ぶ)に行うことができる処理を定義する。ノードは1ステップに次の動作を行う。

- (1) 隣接ノードから、またはローカルで発生したパケットの受信
- (2) 待ちキューにあるパケットの送信
- (3) 移動(リンクの接続, 切断)

概念図を図3に示す。隣接ノードから受信した、あるいは自身が生成した(=ローカルから到着した)パケットは、全て待ちキューに蓄えられる。そして、キューの先頭にあるパケットが、受信、あるいはNext hopへと送信される。ただし、AntNetやOSPFは手法として更新パケットを優先的に送信する。以降では、特に断りが無い限り、メッセージパケットの生成は待ちキューのパケットの有無に関わらず5%の確率で発生し、宛先はランダムに決められるものとする。

5.2 処理するメッセージ数に関して

待ちキューのメッセージの有無に関わらず、全ての

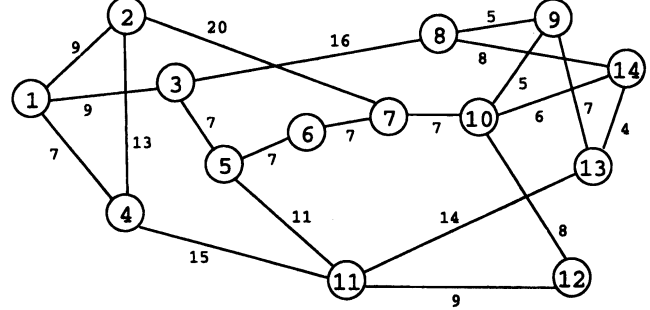


図4 NSFNET とリンク伝播遅延時間 (ms)

ノードが一定確率でメッセージを生成する場合、ルーティングの手法に関わらず、生成されるメッセージパケットの数はほぼ等しくなる。しかし、AntNetの場合、メッセージパケットとは別に、定期的に更新パケット(forward ant)を発行するため、他の手法と比べて処理しなければならないパケット数が多くなる。これは結果としてメッセージパケットの生成確率が高くなったことと同じ影響をもたらす。

そこで、AntNetのみ、forward antが送信された直後は一定時間メッセージパケットの生成を停止することにする。これにより、生成されるメッセージパケットの総数は他の手法に比べて減少するものの、forward antの数を合わせれば処理するパケットの数はほぼ同数となり、ルーティングの性能を平等に評価することができると思う。

5.3 NSFNETにおける実験

まず初めに、図4に示すNSFNET [7] [6]において実験を行った。ただし、伝播遅延時間は考慮せず、全て1ステップで送信できるものとした。

この実験では、ネットワークは固定的で、10000ステップまではリンクの接続や切断は起こらないものとする。10000ステップ実行後、ノード2とノード7を繋ぐリンクを切断し、もう10000ステップ実行する。このときの平均到達ステップ数、学習後の平均到達ステップ数、ネットワーク内パケット数を計測し、比較・評価を行った。

ここで、平均到達ステップ数とは、全てのメッセージパケットの、生成されてから宛先ノードに到達するまでにかかったステップ数の平均値であり、学習後の平均到達ステップ数とは、5000ステップ実行後から計測した平均到達ステップ数である。また、ネットワーク内パケット数とは、生成されてからまだ宛先ノードに到達していないパケットの総数である。

実験に関するパラメータを表1に示す。各手法における内部パラメータは、予備実験を行い、各手法

ノード数	14
シミュレーション	(10000+10000) ステップを 100 回試行
メッセージ生成確率	5 %
OSPF	トポロジ変化時のみ更新パケット発行
AntNet	200 ステップ毎に forward ant 発行, 探求確率 0.002
Ant routing	学習率 6.0
Acc.Ant	学習率 6.0, 2-step backward
提案手法	学習率 6.0, 履歴数 64, 割引率 0.9
上限付き提案手法	同上, 上限値 0.9

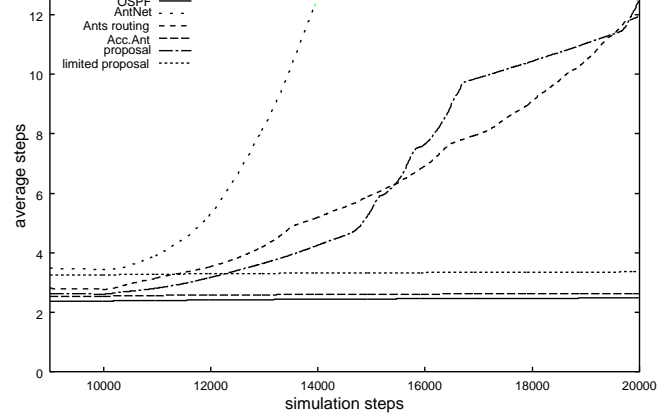


図 7 リンク切断後の平均到達ステップ数

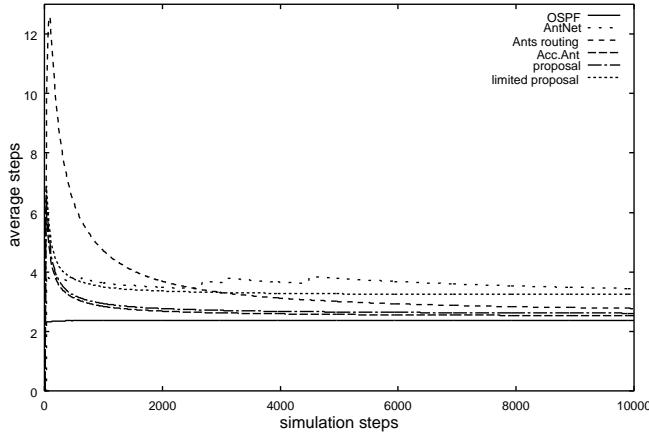


図 5 平均到達ステップ数 (NSFNET)

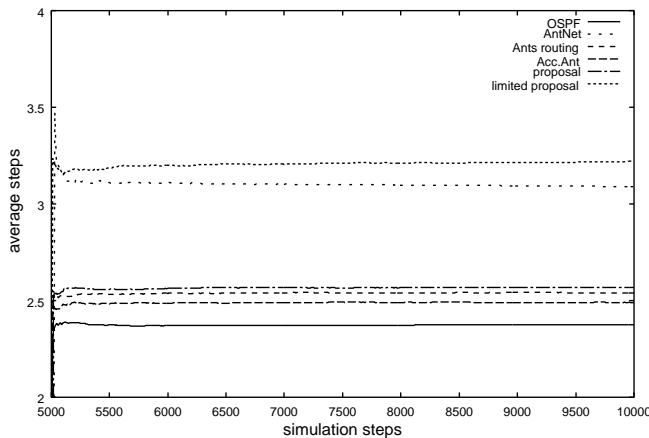


図 6 学習後の平均到達ステップ数 (NSFNET)

に適した値を選択した。

5.3.1 トポロジ変化前

まず, 10000 ステップまでの, 平均到達ステップ数と 5000 ステップ学習した後の平均到達ステップ数の比較を図 5, 図 6 に示す. グラフは 100 回の試行の平均値であり, 横軸はシミュレーションステップである.

まず, 図 5 について, Ants routing がもっとも収束が遅く (6000 ステップ付近), 上限付き提案手法, 提案手法, Acc.Ant の 3 手法がほぼ同時 (2000 ステッ

プ付近) に一定値に収束している. また, AntNet は初めから一定値に収束しているように見えるが, これは 1 ステップ目に全ノードが forward ant を送信するため, ノード数 14 という小さいネットワークでは十分な学習をはじめに行うことができたためと考えられる. このような小さく, バランスの良いネットワークにおいては学習速度に差が出にくい, Ants routing の学習速度が遅いということが言える.

次に, 図 6 について, OSPF が最も短い経路を選択しており, Acc.Ant, Ants routing, 提案手法, AntNet, 上限付き提案手法の順で経路選択の効率が悪くなっていくという結果となった. これは, 確率的ルーティングは経路選択の際にランダム性を含むため, 最適な経路を選択し続けることが難しいためである. AntNet と上限付き提案手法は確率値が 1 に近づきにくい, より大きなステップ数に収束している (注1).

5.3.2 トポロジ変化後

10000 ステップでリンク (2,7) を切断した後の平均到達ステップ数を図 7 に示す. AntNet, Ants routing, 提案手法の 3 つが大きく発散していることがわかる. これは 10000 ステップの十分な学習をさせた後のリンク切断によりブロッキング問題が発生したためである. この結果から, この 3 手法は Acc.Ant や上限つき提案手法と比べ, ブロッキング問題に比較的弱いということが言える.

さらに解析した結果, 100 回の試行のうち, Ants routing と提案手法は共に 20 回, AntNet は 82 回ブロッキング問題が生じていることがわかった. また,

(注1) : AntNet は原論文では OSPF よりも良好な結果を示しているが, これは AntNet があらかじめ十分な学習を行った上で比較を行っているためである. また, トラフィックパターンやパケットサイズ, 伝送遅延等, 実験環境も大きく異なる.

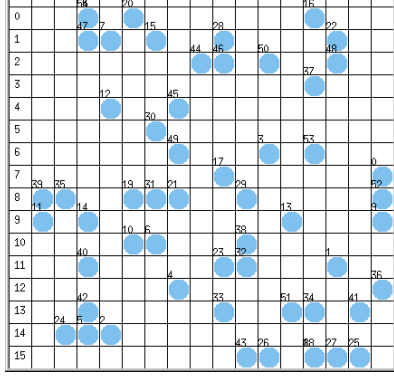


図 8 格子状のフィールド

Acc.Ant と上限つき提案手法はブロッキング問題が全く発生しない、もしくは発生していても即座に対応し確認できないほどであった。ただし、Ants routing と提案手法は 20 回中 17 回が完全にルーティングが破綻していたのに対し、AntNet は完全に破綻したのは 82 回中 7 回で、残りの 75 回は数千ステップ後に元の水準に収束していった。このことから、Acc.Ant、上限つき提案手法は routing-lock に対して頑健であり、AntNet はトポロジの変化に対する即応性に劣るということが言える。

5.4 格子状のフィールドにおける実験

次に、図 8 に示すような 16×16 の格子状のフィールド内を、56 個のノードがランダムに移動する環境において実験を行った。

5.4.1 実験環境

まず、各ノードをランダムに配置し、ユークリッド距離で 3 以内にあるノード同士を通信範囲内にあるものとしてリンクを確立する。

ノードの移動は毎ステップ一定確率で発生し、ランダムに選択された 1 個のノードが上下左右のいずれかへ 1 マスだけ移動する。その結果、通信範囲から外に出たノードのリンクは切断され、通信範囲内に入ったノードのリンクが確立される。このノードの移動が発生する確率を Mobilization Occurrence rate(MO) と定義する。ノードがフィールドの外へ出ることはなく、全体数は常に一定である。

本実験では経路選択の効率性を評価することを目的としているため、ネットワーク全体が 2 つ以上に分割されるようなノードの初期配置・移動は行わないものとする。つまり、全てのノードはいずれかの経路で通信を行うことができ、パケットを伝送することができないノードは発生しない。

MO が低い環境では、ルーティングの結果がネットワークの形状に大きく依存してしまうため、今回

ノード数	56
シミュレーション	50000 ステップを 100 回試行
メッセージ生成確率	5 %
通信可能距離	3(ユークリッド距離)
MO	0.0, 0.0001
OSPF	トポロジ変化時のみ更新パケット発行
AntNet	200 ステップ毎に forward ant 発行、探求確率 0.002
Ant routing	学習率 64
Acc.Ant	学習率 64, 2-step backward
提案手法	学習率 6.0, 履歴数 64, 割引率 0.9
上限付き提案手法	同上, 上限値 0.9

は以下の要素を各手法において同一のものとし、実験を行った。

- (1) ノードの初期配置
- (2) ノードの移動 (発生ステップ, 選択されるノード, 移動する方向)
- (3) メッセージパケットの発生 (発生ステップ, 生成するノード, 宛先)

これらの要素を 1 セットにしたものを 100 パターン用意し、それぞれの手法で実験を行った。実験に関連するパラメータを表 2 に示す。前節同様、各手法における内部パラメータは、予備実験を行い、各手法に適した値を選択した。

予備実験に際し、ノード数が NSFNET に比べて増加したことに伴い、Ants routing や Acc.Ant は学習率の値を増加させる必要があった。これは、この 2 手法がパケットの送信元からの経過時間を元に学習の変化量を決定するためである。これに対し、提案手法 (上限付き提案手法) は送信元からの経過時間に依存せず、近距離のノードに対する学習を重点的に行うため、学習率をそれほど変化させなくとも十分な学習量を得ることができる。このことから、提案手法は内部パラメータにおいて頑健であるということが言える。

5.4.2 MO=0.0 における実験

まずは MO=0.0, つまり初期配置からノードの移動が起こらない環境での実験である。NSFNET と比べてノード数が 4 倍になっており、各ノードごとのリンク数の差も大きくなるため、ルーティングは困難となる。

実験の結果、安定した状態でのネットワーク内パケット数から 10 倍以上パケット数が増加した状態が 3000 ステップ続いた場合、ルーティングが破綻したものとし、経路が破綻せずに正しく経路を獲得でき

手法	66 回中収束した回数 (%)
OSPF	35 回 (53.0 %)
AntNet	37 回 (56.1 %)
Ant routing	54 回 (81.8 %)
Acc.Ant	62 回 (93.9 %)
提案手法	62 回 (93.9 %)
上限付き提案手法	59 回 (89.4 %)

た回数を表 3 に示す。全手法に共通して破綻している 34 回のパターンはノードの初期配置からボトルネックがあったもので、評価の対象からは外し、残りの 66 回のうち、収束した割合を併記した。

固定的なネットワークにも関わらず、OSPF の収束率ももっとも低い結果となった。OSPF は各ノードが自身を根とした spanning tree を作成し、最短コストの経路を選択する手法であるが、経路の節点 (ノード) の処理遅延は考慮に入れておらず、一般的に輻輳制御はルータの実装に委ねられる。そのため、今回のモデル化した実験環境では、最短ホップの経路上のノードにおいて輻輳が生じた場合、OSPF はこれを回避することができない。これにより収束率が低下したと思われる。

また、他の強化学習的手法でいくつか破綻しているが、これは経路の学習が進む前にネットワーク内のパケット数が増大したことが原因である。特に AntNet は他の手法と比較して学習速度が遅いと言える。

100 回中、全ての手法で収束した 27 回の平均値を算出した、平均到達ステップ数と学習後の平均到達ステップ数を図 9, 10 に示す。学習後の平均到達ステップ数は 20000 ステップ実行後から計測したものである。

図 9 について、提案手法がもっとも学習が早く、上限付き提案手法、Acc.Ant、Ant routing、AntNet の順に一定値への収束が遅くなっている。OSPF は手法が異なるため収束という概念はなく、シミュレーション開始時から一定値である。また、図 10 について、学習後は提案手法がもっとも最適に近い経路を選択しており、Acc.Ant、OSPF、上限付き提案手法、Ant routing、AntNet の順に選択する経路の効率が悪くなっている。

固定的なネットワークにも関わらず、最短経路を選択する OSPF が提案手法や Acc.Ant よりも悪い結果となっているのは、前述の通り、OSPF は輻輳回避を行わないためである。提案手法や Acc.Ant はノードの処理遅延を考慮したルーティングを行うため、OSPF と比べ効率の良い経路を獲得できる。ま

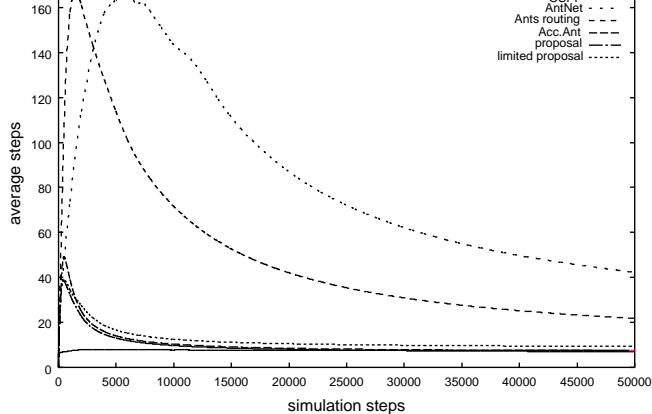


図 9 平均到達ステップ数 (MO=0.0)

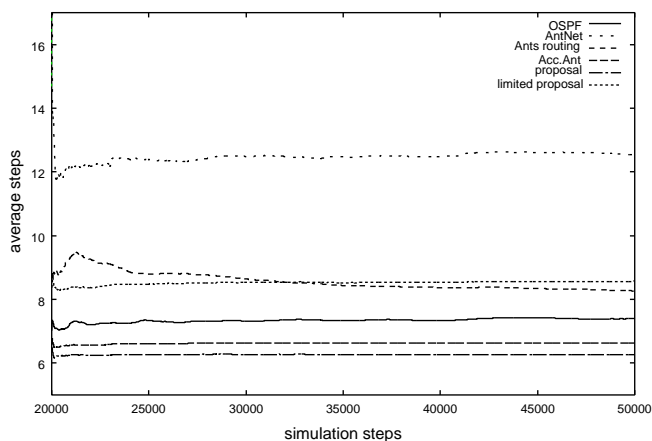


図 10 学習後の平均到達ステップ数 (MO=0.0)

た、AntNet や Ants routing は、50000 ステップ以降、更に最適な経路に近づいていくものと思われるが、提案手法と比べて明らかに学習が遅いと言える。上限付き提案手法は前節と同様、学習は速いものの、静的な環境では最適な経路に近づきにくいことが確認された。

5.4.3 MO=0.0001 における実験

次に、MO=0.0001 の環境で実験を行った。前節と同様、全手法に共通して破綻した 45 回の試行はノードの初期配置やトポロジの変化からボトルネックが生じたものであり、評価の対象からは外す。残りの 55 回のうち、各手法ごとに収束している回数とそのパーセンテージ、routing-lock が原因で収束しなかった回数を表 4 に示す。

OSPF は他の手法の半分以上しか収束せず、もはやこの環境では実用的ではない。強化学習的手法では、55 回の試行のうち 18 回の試行で routing-lock が発生したが、上限付き提案手法はそのうち 1 回しか破綻しておらず、全体の収束率も最も高い。このことから、上限付き提案手法は他の手法と比べて頑健

手法	55 回中収束した回数 (%)
OSPF	26 回 (47.3 %)
AntNet	39 回 (70.9 %), routing-lock:11 回
Ants routing	47 回 (85.5 %), routing-lock:8 回
Acc.Ant	52 回 (94.5 %), routing-lock:3 回
提案手法	48 回 (87.3 %), routing-lock:7 回
上限付き提案手法	54 回 (98.2 %), routing-lock:1 回

なルーティングが可能であるということが言える。

Acc.Ant も routing-lock による破綻が 3 回と比較的高い頑健性を示しているが、これは No return rule によるものと思われる。Acc.Ant は直前のノードを選択候補から除外するが、他のノードを選択する確率値が非常に小さい場合には No return rule を無視し、接続ノードの中からランダムに Next hop を選択する。これは袋小路の経路に入ってしまった場合に、パケットを宛先まで送信できなくなることを防ぐためである。そのため、routing-lock が発生した際もランダムに経路を探索する可能性が高くなり、結果的にブロッキング問題を回避しているものと考えられる。

100 回中、全ての手法で収束した 20 回の試行の平均値を算出したものが図 11, 図 12 である。

傾向は MO=0.0 のときと変わらず、提案手法が最も速く最適に近い経路に収束している。

6. まとめと今後

提案手法は従来手法と比べ、学習速度が速く、最適に近い経路を獲得できることを確認した。また、上限付き提案手法は最適な経路を獲得することは難しいものの、学習速度は提案手法と同等であり、routing-lock に対して頑健なルーティングが可能であることも同様に確認した。

上限付き提案手法は一定値で選択確率を更新することを止めてしまうため、ルーティングに無駄な経路が含まれてしまうことが避けられない。一方、提案手法は確率値を 1 まで更新することができるため、最適に近い経路を選択し続けることが可能であるが、routing-lock が発生する可能性が高い。

上限つき提案手法は routing-lock に対してある程度有効であるが、根本的な解決には至っていない。今後は、より routing-lock に対して頑健なルーティングアルゴリズムの検討と、さらに詳細な実験を行い、性能の評価を行う予定である。

文 献

[1] IETF MANET Working Group
<http://www.ietf.org/html.charters/manet-charter.html>

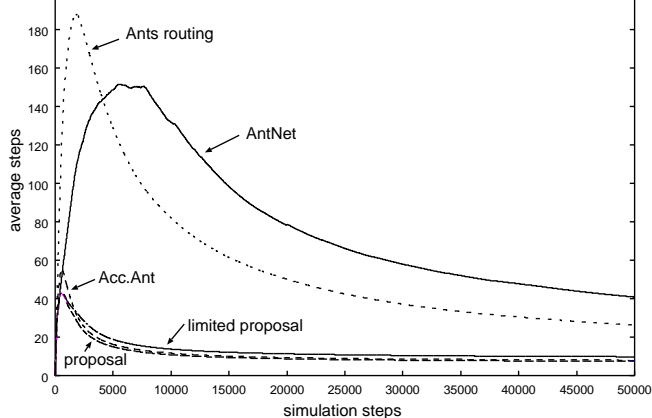


図 11 平均到達ステップ数 (MO=0.0001)

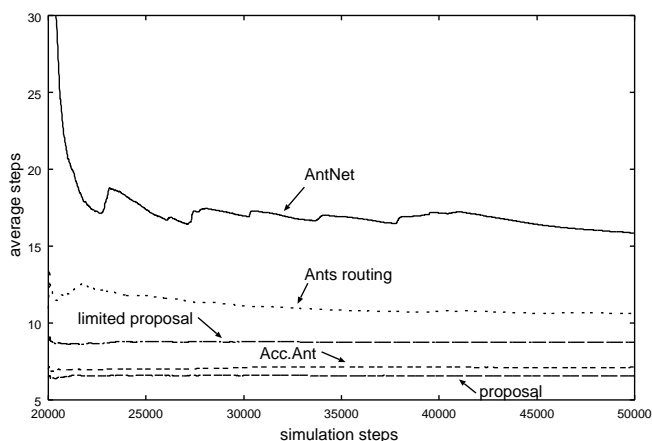


図 12 学習後の平均到達ステップ数 (MO=0.0001)

[2] M.Littman, J. Boyan: A Distributed Reinforcement Learning Scheme for Network Routing Proceedings of the 1993 International Workshop on Applications of Neural Networks to Telecommunications (1993)

[3] S.Kumar, R.Miikkulainen: Dual Reinforcement Q-routing: An On-line Adaptive Routing Algorithm Proceedings of the Artificial Neural Networks in Engineering Conference (1997)

[4] D.Subramanian, P.Druschel, J.Chen: Ants and Reinforcement Learning: A Case Study in Routing in Dynamic Networks Proceedings of IJCAI-97 (1997)

[5] Hiroshi Matsuo, Kouichi Mori: Accelerated Ants Routing in Dynamic Networks 2nd International Conf. on Software Engineering, Artificial Intelligence, Networking & Parallel/Distributed Computing, accepted (2001)

[6] 種田和正, 片岡明: ロックフリー AntNet とその適応能力の評価 電子情報通信学会論文誌 B Vol. J82-B No.7 (1999)

[7] Gianni Di Caro, Marco Dorigo: AntNet: Distributed Stigmergetic Control for Communications Networks Journal of Artificial Intelligence Research 9 (1998)