

アドホックネットワークにおけるネットワークの絞り込みのための 効率的なトポロジー検出アルゴリズム

大庭 将平†, バロリ レオナルド‡, 池田 誠†
デマルコ ジュゼッペ‡, 小山 明夫†‡

近年、携帯電話・PDA・ノートパソコンの爆発的な普及により無線ネットワークへの需要が高まっている。特にインフラストラクチャを必要としない無線アドホックネットワークは、無線基地局の設置コストの削減などの点から注目されている。しかし無線アドホックネットワークには、端末移動・入退出に伴い激しくネットワークのトポロジーの変化してしまうなどの問題が挙げられる。以前の研究では、無線アドホックネットワークのための遺伝的アルゴリズムベースのルーティング方法を提案した。本稿で遺伝的アルゴリズムベースのルーティングアルゴリズムの探索スペースを減少させるために、効率的なトポロジー検出アルゴリズムを加えることにより、遺伝的処理の計算量の短縮を目指し、効率的かつQoSを確保したルーティングを目指す。

An Effective Topology Extraction Algorithm for Search Reduction Space in Ad-hoc Networks

Shohei Ohba †, Leonard Barolli ‡, Makoto Ikeda †
Giuseppe De Marco ‡, Akio Koyama † ‡

† Graduate School of Engineering, Fukuoka Institute of Technology

E-mail: {mgm05004, mgm05001}@ws.ipc.fit.ac.jp

‡ Faculty of Information Engineering, Fukuoka Institute of Technology

E-mail: {barolli, demarco}@fit.ac.jp

† ‡ Faculty of Engineering, Yamagata University

E-mail: akoyama@yz.yamagata-u.ac.jp

A lot of work has been done on routing in Mobile Ad-hoc NETWORKS (MANETs). However, the proposed routing solutions deal only with the best effort data traffic. Connections with Quality of Service (QoS) requirements are not supported. The QoS routing has been receiving increasingly intensive attention in the wireline network domain. However, for MANETs are proposed only few QoS routing algorithms. For MANETs, approximated solutions and heuristic algorithms should be developed and QoS routing algorithms should be adaptive, flexible, and intelligent. In our previous work, we proposed a Genetic Algorithm (GA) based routing method for MANETs. In this paper, we enhance the proposed algorithm by adding an effective topology extraction algorithm to reduce the search space of the GA-based routing algorithm. The performance evaluation via simulations show that proposed search space algorithm has a good performance.

1. はじめに

無線アドホックネットワーク[1]-[2]とは、移動端末同士が一時的に構築する自律型無線ネットワークである。従来の無線ネットワークでは、端末がネットワークへアクセスする方法として、セル方式の移動通信網においては基地局、無線 LAN においてはアクセスポイントなどのネットワーク設備が必要になる。また端末同士が通信を緒かなう場合でも基地局を経由する必要があり基地局が存在しない環境ではネットワークを構築することができない。このように従来の通信ネットワークはあらかじめインフラストラクチャを用いて端

末間の通信を行うことが基本となっている。これに対し無線アドホックネットワークでは、移動端末同士が互いに無線通信を行い、それぞれの端末がルーティングを行うので、ケーブルや基地局などのインフラストラクチャを用いずに、ネットワークを構築できる。またノードがある程度広い範囲に分配配置している場合やノード間に遮蔽物が存在する場合任意の始点ノードから終点ノードへ直接通信（ワンホップ通信）行えるとは限らない。そのような場合には途中に存在する中継ノードを利用し、マルチホップ通信を行うことにより実現できる。つまり端末として機能するだ

けでなくルーティング機能も持ち合わせている。しかし基地局のような中央制御的な役割を果たす設備が無く個々の端末が自由に移動するので、通信相手までの経路選択が重要な問題となる。また、ネットワークに関する情報を各ノードが分散して持っている場合、通信経路を動的に変化させるのは非常に難しい。しかし、あるノードでネットワーク全体の状態を把握し、そのノードにおいて目的地までのルーティングを行えば、通信経路を動的に変化させることがたやすくなる。そこで、送信元のノードにおいて目的地までの通信経路を決定する始点制御ルーティング[3]を用いた経路制御を行う。また、通信経路を動的に変化させるために遺伝的アルゴリズム[4]-[5]を用い、ネットワーク環境に適した進化的な経路選択を行う。

2. 遺伝的アルゴリズムの処理手順

本研究では、QoS[6]を確保した経路を求めるのに遺伝的アルゴリズム (Genetic Algorithm: GA) を用いる。GA とは、選択や交叉、突然変異といった遺伝的操作を各個体に適用し、遺伝的要素の組み換えを行うことによって次の世代の個体を作り、個体集団を進化させる。その手順は次のようになる。

まず初めに、初期世代を生成する。一般には、決められた個体数の染色体を遺伝子プールにランダムに生成する。そして、この初期集団の各々の個体に対して適応度が決定されたら、その適応度により個体の選択確率を設定する。基本的には、適応度の高い個体が生き残り、より多くの子孫を残す機能となるため、適応度が高くなるほど選択確率も高くなるように設定する。選択確率が決定されたら、その値を基に遺伝子プールの中から交叉する個体を選択する方法はいくつもある。どの選択方法を用いるか、さらに適応度と残せる子孫の数の期待値の関係をどのように設定するかで、淘汰確率は変わってくる。

交叉を行う個体対が決定したら、適当な確率で染色体の交叉を行う。基本的には、選択された2個体の双方の染色体の一部ずつを採ってきて子孫の染色体を作る。普通は新しい個体(子)の染色体の*i*番目の遺伝子座には、どちらかの親の染色体*i*番目の遺伝子座の遺伝子を複製する。このような交叉が終わると新しい個体がつくられたことになる。

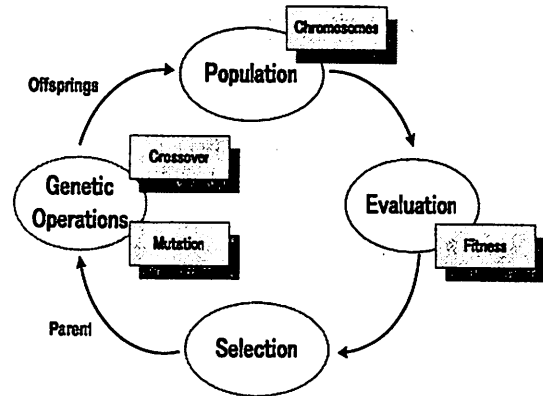


図1 遺伝的アルゴリズムの処理手順

これらの操作をあらかじめ競ってした個体数を満たすまで繰り返す。個体数が満たされたら、次に突然変異を加える。突然変異もまた交叉と同様、適当な確率で行う。突然変異が終わると、新しい個体集合が生成されたことになる。新しい世代の個体集合が生成されたら、古い世代を新しい世代に置き換える。そして、この新しい個体集団に足して、あらかじめ与えられた条件(例えば世代数や適応度の設定など)を満たすまで、再び、適応度の評価、交叉、突然変異を繰り返す。以上の処理の様子を図1で示す。

3. 従来の研究

従来の研究である LBR (Load Balancing Routing) [7]ではネットワークをそのままの形で使用し、通信経路のノードの並び替えがそのまま遺伝子として表現されている。この方法では、遺伝的操作を実行して問題を解く際、必要のない場所を探索したり、探索すべき場所を探索しなかったりすることがある。この問題点を改善したものが TMR (Tree Model Routing) [8]-[9]である。TMRでは遺伝的操作を行いやすくし、無駄な計算を行わなくて済むように、ネットワークをそのままの形で使用するのではなくツリー表現のネットワークを用いて遺伝子のマッピングを行う。しかし両場合ともノード数が増加する場合、遺伝的操作が複雑になり計算量が増加してしまう問題点がある。従来ルーティング方法の処理手順として、無駄な計算を行わなくて済むように、無効グラフのネットワークをそのまま使用するのではなく、ツリー表現のネットワークを用いて遺伝子のマッピングを行っている。図2のネットネットワークにおい

てツリー表現法を示す。まず出発点を「A」、目的値を「H」とした時のツリー表現は図3のようになる。ここでツリーの作成方法を説明する。最初に図2の出発点であるノード「A」を親とし、ノード「A」に隣接する「B」、「C」、「E」が、ノード「A」の子供となるようにツリーを作成する。同様に、「B」を親とする子供「D」、「E」を作成する。以下、同様に同じ作業を繰り返すことでノード「A」を頂点とするツリーを作成することができる。このとき、ループが発生しないようにすべての経路を作成する。次に、このツリーの重複する分岐点(図3の網掛け部)をまとめて、ツリーを新たに作成する。この時重複する部分とは、分岐点のノードが同じであり、その親のノードも同じであり、かつその子供のリンクがすべて一致しているものを指す。一つでもリンクが一致していない場合、まとめることができない。その後、ツリーの各分岐点に番号をつける(図4)。階層が浅い順に若い番号が設定されるように番号付けが行われている。このように作成されたツリーを用いて、ネットワーク経路を表す個体を表現する。ここで、各分岐点を遺伝子とした時、その並びによって示される染色体から経路が決定される。ここでいう遺伝子とは表1にあるような配列番号でとることができる値を指し、染色体とはその並びによって示される一つの配列と考えて頂きたい。例えば「A-B-D-E-C-F-G-H」を選んだ場合の配列は表2のようになる。このとき、配列内で使用している場所には、使用していることを示すフラグを立てておく。こうして得られた個体は隣接する遺伝子同士が相互作用しない。よってルーティングを行う最、突然変異や交叉といった遺伝的操作が施しやすい。さらに、重複部分をまとめているため、個体の染色体の遺伝子が少なくなり、ルーティングを行う際の計算量を削減することができる。

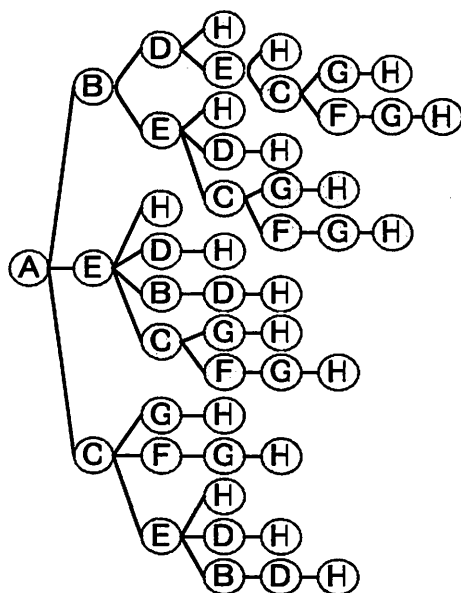


図3 ツリーモデルI

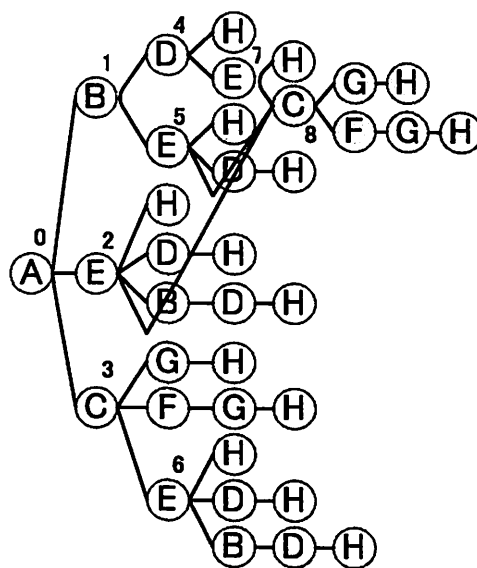


図4 ツリーモデルII

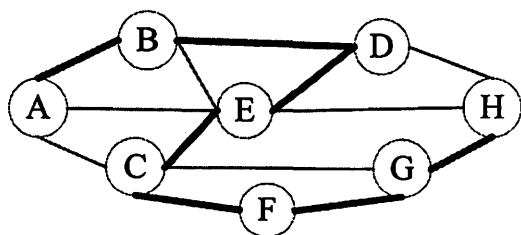


図2 ノード数8のネットワークモデル

表1 遺伝子コーディングI

0	1	2	3	4	5	6	7	8
B	D	H	G	H	H	H	H	G
E	E	D	F	E	D	D	C	F
C		B	E		C	B		
		C						

表 2 遺伝子コーディング II

0	1	2	3	4	5	6	7	8
B	D	-	-	E	-	-	C	F
1	1	0	0	1	0	0	1	1

4. 提案手法

ネットワークのノード数が増加するたびに遺伝的処理にかかる計算時間が増大してしまう。その計算時間を削減するために、あらかじめネットワークの絞込みを行い、遺伝的処理の計算量の短縮を目指し、効率的かつ QoS を確保したルーティングを行う。また提案手法のフローチャートを図 5 に示す。具体的な処理手順として、各リンクにおいてエラー率、各ノードにおいてバッファの値を設定し、閾に満たなければ除外し満たしていればそれを含め、最終的に残ったものが新しいトポロジーとなる。

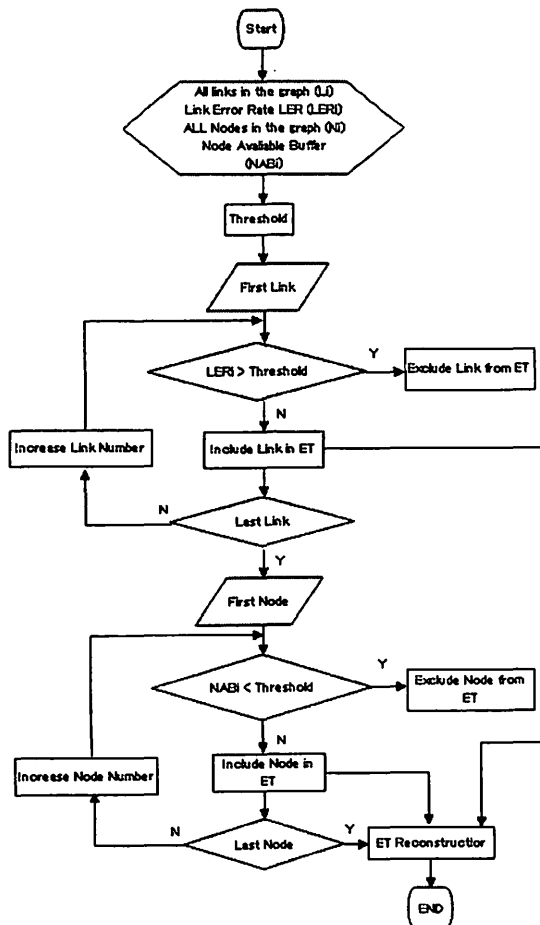


図 5 提案手法フローチャート

5. 評価実験

この章は、4 章で提案したシステムについて行った評価実験の方法、結果と、その考察について述べる。評価実験を行うために使ったシミュレーション環境としては CPU Intel Pentium M1.3GHz, Memory 768MB, OS Microsoft Windows XP Pro, 開発環境として Microsoft Visual C++6.0 で実験を行った。

5.1 実験方法

本アルゴリズムの性能を評価するため、ノード数 20, 30, 40, 50 の 4 パターンのノードで実験を行った。まずランダムにリンクを作りトポロジーを形成する。次にパラメータとして、バッファの残り容量、エラー率を設定する。エラー率は 0.01~1.0%, バッファの残り容量は 0~100% にランダムに設定した。また各パラメータに対する閾値としてエラー率は全リンクのエラー率の平均を基準として 0.5~1.5 まで 0.25 きざみで掛け合わせたものとし、バッファの残り容量は全ノードのバッファの残り容量の平均を基準とし 0.5~1.5 まで 0.25 きざみで掛け合わせたものにした。閾を設定した場合と、しなかった場合で、ソースノードからデスティネーションノードまでの全経路を探索するのにかかる平均処理時間と平均リンク数を比較した。また平均を取るために各閾値ずつ 300 回実験を行った。

5.2 実験結果

アルゴリズムの集計結果は以下の通りである。閾を設定した場合の結果を図 6, 図 7 に示す。図 6 は各ノード数における処理時間とリンク数、また図 7 は各ノード数に対する処理時間と探索成功率である。またパラメータは全リンク中のエラー率の平均×0.75, 全ノードのバッファの残り容量の平均×1.25 に設定したものである。閾値を高め設定した場合処理時間を削減することはできているが、リンク数は 0 に近い値を示しており、成功率は 50%前後の値を示している。図 8, 図 9 は閾値として全エラー率の平均×1.25, 全バッファの残り容量の平均×0.75 に設定したものである。この場合、処理時間も大幅に削減できている。また成功率はほぼ 100%に近い値を示しており、良好な結果が得られたと言える。リンク数においても閾を設定しなかった場合と比べて大幅に削減できたと言える。

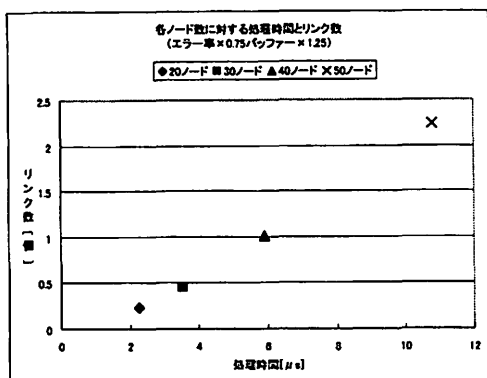


図6 ノード数の変化による処理時間とリンク数 I

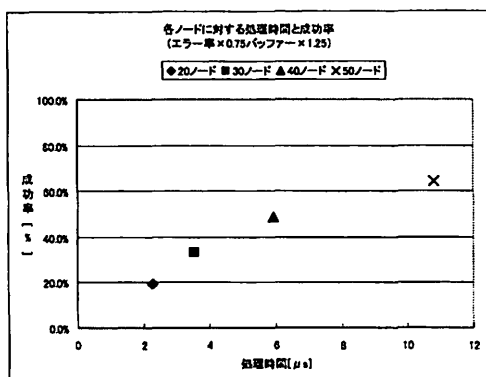


図7 ノード数の変化による処理時間と成功率 I

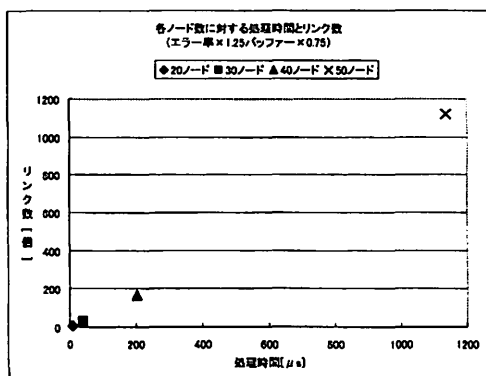


図8 ノード数の変化による処理時間とリンク数 II

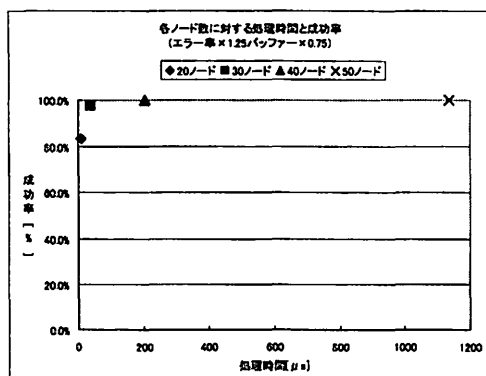


図9 ノード数の変化による処理時間と成功率 II

6. まとめ

今回の提案手法を用いることにより従来からの手法をさらに効率よく行える有効性が示されたといえる。しかし図7, 図8の様に閾を高く設定するとルーティングを行うためのリンク数が減少しトポロジーを検出することができないことがわかる。ノード数に対する閾値の値を考慮し、動的に設定できるようにしなければならないことが伺える。

今後の研究として、無線アドホックネットワークには多種類の端末が存在するため、端末のバッテリー容量も様々である。無線アドホックネットワークでは、各ノードが他のノードからのパケットを転送することによって実現されるので、電力消費は重要な問題となってくる。今後は電力を考慮したパラメーターを使いルーティングを行う。

参考文献

- [1] C.K. Toh 著, 構造計画研究所 訳, “アドホックワイヤレスネットワークプロトコルとシステム”, 共立出版, 2003.
- [2] 小牧 省三, 間瀬 憲一, 松江 英明, 守倉 正博, “無線 LAN とユビキタスネットワーク” 丸善, 2004.
- [3] 村井 康紀, 棟朝 雅晴, 高井 昌彰, “遺伝的アルゴリズムによる始点制御ルーティングの一手法”, 情報処理学会研究報告, Vol.8, No.3, pp.312-319, 1993.
- [4] 北野 宏明, “遺伝的アルゴリズム④”, 産業図書, 2000.
- [5] 北野 宏明, “遺伝的アルゴリズム”, 人工知能学会誌, Vol.7, No.1, pp.26-37, 1992.
- [6] 河野 圭太, 益田 智員, 木下 和彦, 村上 孝三, “複数の QoS を考慮したサービス情報検索方式”, 電子情報通信学会論文誌, Vol.J84-B, No.3, pp.443-451, 2001.
- [7] 棟朝 雅晴, 高井 昌彰, 佐藤 義治, “負荷分散機構を有する適応型ルーティングの一手法”, 情報処理学会研究報告, Vol.97, No.13, pp.205-210, 1997.
- [8] L. Barolli, A. Koyama, T. Suganuma, N. Shiratori, “A GA Based QoS Routing Method for Mobile Ad-Hoc Networks”, JOIN, Vol.4, No.3, pp.251-270, 2003.
- [9] K. Matsumoto, A. Koyama, L. Barolli, Z. Cheng, “A QoS Routing Method for High-Speed Networks Using Genetic Algorithm”, IPSJ Journal, Vol.42, No.12, pp.3121-3129, 2001.