A-017

単位円グラフの最小支配集合問題に対する局所並列近似アルゴリズムの提案

宇野　拓也，山田　敏規
埼玉大学大学院 理工学研究科 数理電子情報系専攻

1　導入

無線アドホックネットワークは，多数の端末が相互に接続するマルチホップ通信技術を用いているため，固定インフラがない領域でも他の端末を中継することで通信エリアを拡大することができる特徴をもつ．しかし，動的に変化するネットワークでのルーティング，通信における出力の大きさなどの調節は各端末 (ノード) 自身が行わなければならない．つまりノードは分散アルゴリズムを適用することにより，通信サービスを構築する必要がある [8].

トポロジーが常時変化しうる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の性能を向上させる方法の一つとして，ノードをクラスタでまとめるというものがある．この方法ではルーティングはルータとして機能するクラスタヘッドの間で行われ，他のノードは隣接するクラスタヘッドを介して通信を行う．そのため効率的な通信を可能にするためには，全てのノードの近傍に一つのクラスタヘッドが必要である．無線アドホックネットワークにおける他のクラスタリング分散アルゴリズムは文献 [3–5](一般グラフ) や文献 [2,7,9](制約付きグラフ) で述べられている．いくつかのクラスタリングアルゴリズムの実験的比較は [1] で行われている．

上記のクラスタヘッドはグラフの支配集合に対応する点集合 $V(G)$，辺集合 $E(G)$ のグラフ G を考える．グラフ G の支配集合 S とは任意の点 $v \in V(G) - S$ が S 内に少なくとも一つの近傍をもつ，という条件を満たす部分集合 $S \subseteq V(G)$ である．支配集合の定義は支配集合内のノード同士の隣接の可能性を含むが，無線アドホックネットワークの性質を考慮した場合，クラスタヘッドを隣接させることは望ましくない場合がある．この支配集合のノード同士を隣接させないという追加要件は，グラフの極大独立集合の概念につながる [8]．グラフ G の独立集合 $S \subseteq V(G)$ は，任意の二つのノード $u, v \in S$ が G で隣接しないようなノードの部分集合である．このとき S が支配集合を形成するなら，S は極大独立集合 (**MIS**) である．本研究で求める支配集合はこの極大独立集合である． アドホックネットワークのモデルには単位円グラフを用いる．これはノードを頂点で表し，通信範囲は単位円として扱い，ノード間距離が 1 以内であれば隣接しているとして辺で結ぶことで表現される．

極大独立集合を求めるために局所並列アルゴリズムを使用する．単純な局所並列アルゴリズムとして **Greedy** 局所並列アルゴリズムが存在する．これは実験的に少ない計算時間で良い解を与えるという点で優秀なアルゴリズムではあるが，最悪時の計算時間が $n/2$ であり，時間計算量は $O(n)$ である．**Luby** は確率を取り入れた局所並列アルゴリズムを提案しており，これは高確率で時間 $O(\log n)$ で解を与えるアルゴリズムとなっている [6]．ただ，**Luby** のアルゴリズムの実験的計算時間は **Greedy** 局所並列アルゴリズムのものと比べ大きくなるという欠点がある．本研究では **Greedy** 局所並列アルゴリズムをベースとし，最悪時での計算時間の増大を解消するような局所並列アルゴリズムを提案する．

2　アルゴリズムの概要

2.1　Greedy 局所並列アルゴリズム

全てのノードを **active** にし，全てのノードが **MIS** の要素か **passive** ノードになるまで以下の操作を繰り返す．

1. 各ノード v が **active** であるとき，隣接する **active** ノードと **ID** を比較し，自身の **ID** が最も大きい (小さい) とき v を **MIS** に加える．

2. ノード v に隣接するノードの中で，**MIS** に属するノード u が存在するなら，ノードを **passive** にする．

このアルゴリズムにおいてノードが **ID** 順に距離 1 で一列に配置されている場合に最悪となり，この場合一つのラウンドで **MIS** に加えられるノードは最も **ID** の大きい (小さい) 一つだけであり，それに隣接する **active** ノード一つだけが **passive** となるので，全てのノードが **MIS** の要素か **passive** ノードになるまでには $n/2$ ラウンド必要となる．そのため時間計算量は $O(n)$ となる．

2.2　Luby 局所並列アルゴリズム [6]

全てのノードを **active** にする．全てのノードが **MIS** の要素か **passive** ノードになるまで以下の操作を繰り返す．

1. 各 **active** なノード v に対し，自身を $\dfrac{1}{2d(v)}$ の確率でマークする ($d(v)$ はノード v に隣接する **active** ノードの数を示す)．

2. ノード v に隣接する全てのマークされたノード u に対し $d(v) > d(u)$ を満たすなら，v を **MIS** に加える．$d(v) = d(u)$ なら **ID** を比較し，v の方が大きい (小さい) なら **MIS** に加える．

3. ノード v に隣接するノードの中で，**MIS** に属するノード u が存在するなら，ノードを **passive** にする．

2.3　提案アルゴリズム (1)，(2)

全てのノードを **active** にする．全てのノードが **MIS** の要素か **passive** ノードになるまで以下の操作を繰り返す．提案アルゴリズム (1) と (2) では与える乱数値の範囲のみ異なる．

1. 各 **active** なノード v に対し，提案アルゴリズム (1) は $[1, \alpha \cdot d(v)]$ の範囲，提案アルゴリズム (2) は $[\alpha \cdot d(v), \alpha \cdot d(v) + (\alpha - 1)]$ の範囲で乱数値 r_v を与える．α は任意の正の自然数．

2. ノード v に隣接する全ての **active** ノード u に対し $r_v > r_u$ を満たすなら，v を **MIS** に加える．$r_v = r_u$ なら **ID** を比較し，v の方が大きい (小さい) なら **MIS** に加える．

3. ノード v に隣接するノードの中で，**MIS** に属するノード u が存在するなら，ノードを **passive** にする．

各提案アルゴリズムは以下の定理が成立する．

定理 1 **Greedy** アルゴリズムの最悪時のノード配置において，提案アルゴリズム (1) は $1 - \dfrac{1}{n}$ 以上の確率で $O(\log n)$ ラウンドで終了する．

定理 2 **Greedy** アルゴリズムの最悪時のノード配置において，$\alpha \geq 2$ のとき提案アルゴリズム (2) は $1 - \dfrac{1}{n}$ 以上の確率で $O(\log n)$ ラウンドで終了する．

3 計算機実験

3.1 実験方法

実験は以下の条件のもとで行われる．

- 配置するノードの数は 10000 個．
- ノードを配置する空間を $x * x$ の正方形とし，x の値を $x = 100, 75, 50, 25, 10, 5, 4, 3, 2, 1$ と変化させる．
- データはそれぞれ試行回数 100 の平均より算出される．

まず，提案アルゴリズム (1)，(2) の α の値を決定するために予備実験として実験 1，2 を行い，α の値を固定して本実験 3 を行った．

1. 提案アルゴリズム (1) に対し，ノードに割り振る乱数値の範囲である $[1, \alpha \cdot d(v)]$ における α の値を変化させ，それぞれの平均ラウンド数と平均支配ノード数を求める．
2. 提案アルゴリズム (2) に対し，ノードに割り振る乱数値の範囲である $[\alpha \cdot d(v), \alpha \cdot d(v) + (\alpha - 1)]$ における α の値を変化させ，それぞれの平均ラウンド数と平均支配ノード数を求める．
3. **Greedy**，**Luby**，提案アルゴリズム (1)，(2)(各 α は固定) それぞれに対し，平均ラウンド数と平均支配ノード数を求める．

実験環境は以下のとおりである．

CPU	Intel Core i3 3.07GHz
メモリ	3.0GB
OS	Ubuntu 10.04
コンパイラ	GCC(ver 4.4.3)

3.2 実験結果

実験 1 の結果を表 1 に，実験 2 の結果を表 2 に示す．結果は α の値にそれほど依存していないため，実験 3 における各提案アルゴリズムの α は便宜的に 5 とする．実験 3 の結果を表 3，4 に示す．

表 1: 実験 1 結果

	平均ラウンド数			平均支配ノード数		
$x * x$	$\alpha = 1$	$\alpha = 5$	$\alpha = 10$	$\alpha = 1$	$\alpha = 5$	$\alpha = 10$
100*100	3.21	3.15	3.15	3572.88	3575.71	3575.46
50*50	4.42	4.48	4.50	1267.20	1264.20	1262.54
10*10	5.82	5.83	5.89	73.82	73.96	73.50
5.0*5.0	5.21	5.20	5.26	21.33	21.58	21.33
3.0*3.0	4.64	4.71	4.62	9.31	9.39	9.21
1.0*1.0	1.00	1.00	1.00	1.00	1.00	1.00

表 2: 実験 2 結果

	平均ラウンド数			平均支配ノード数		
$x * x$	$\alpha = 1$	$\alpha = 5$	$\alpha = 10$	$\alpha = 1$	$\alpha = 5$	$\alpha = 10$
100*100	3.94	3.89	3.87	3287.45	3287.13	3286.60
50*50	5.79	5.59	5.58	1142.14	1141.99	1141.43
10*10	6.25	6.22	6.23	70.83	71.06	71.13
5.0*5.0	5.44	5.42	5.41	21.26	21.25	21.15
3.0*3.0	4.79	4.80	4.81	9.16	9.15	9.16
1.0*1.0	1.00	1.00	1.00	1.00	1.00	1.00

表 3: 実験 3 結果 [平均ラウンド数]

	平均支配ラウンド数			
$x * x$	**Greedy**	**Luby**	提案 (1)	提案 (2)
100*100	3.61	10.74	3.19	3.91
50*50	4.35	12.83	4.51	5.57
10*10	4.27	13.04	5.90	6.38
5.0*5.0	3.46	11.36	5.29	5.30
3.0*3.0	3.02	8.84	4.65	4.83
1.0*1.0	1.70	4.70	1.00	1.00

表 4: 実験 3 結果 [平均支配ノード数]

	平均支配ノード数			
$x * x$	**Greedy**	**Luby**	提案 (1)	提案 (2)
100*100	3837.80	3876.49	3576.72	3289.20
50*50	1370.61	1382.71	1261.70	1143.64
10*10	75.43	75.81	73.70	70.70
5.0*5.0	21.56	21.98	21.28	20.85
3.0*3.0	9.22	9.19	9.36	9.19
1.0*1.0	1.80	1.77	1.00	1.00

参考文献

[1] S. Basami, M. Mastrogiovanni, and C. Petrioli. *A performance comparison of protocols for clustering and backbone formation in large scale ad hoc networks. In: Proceedings of the 1st IEEE International Conference on Mobile Ad-hoc and Sensor Systems (MASS'04).* IEEE Computer Society Press, 2005.

[2] J. Geo, L. Guibas, J. Hershberger, L. Zhang, and A. Zhu. *Discrete mobile centers. In: Proceedings of the 17th Annual Symposium on on Computational Geometry(SCG'01).* ACM Press, 2001.

[3] L. Jia, R. Rajaraman, and T. Suel. *An efficient distributed algorithm for constructing small dominating sets. In: Proceedings of the 20th ACM Symposium on Principles of Distributed Computing(PODC'01),.* ACM Pres, 2001.

[4] F. Kuhn, T. Moscibroda, and R. Wattenhofer. *The price of being near-sighted. In: Proceedings of the 17th Annual ACM-SIAM Symposium on Discrete Algorithms(SODA'06).* ACM Press, 2006.

[5] F. Kuhn and R. Wattenhofer. *Constant-time distributed dominating set approximation. in: Proceeding of the 22nd ACM Symposium on Principles of Distributed Computing(PODC'03).* ACM Press, 2003.

[6] M. Luby. *A simple parallel algorithm for the maximal independent set problem. In Proceeding of the 17th Annual ACM Symposium on Theory of Computing(STOC'85).* ACM Press,, 1985.

[7] S. Parthasarathy and R. Gandhi. *Distributed algorithms for coloring and connected domination in wireless ad hoc networks. In Lodaya, K., Mahajan, M. (eds.) FSTTCS 2004. LNCS.* Springer, 2004.

[8] D. Wagner and R. Wattenhofer. *Algorithms for Sensor and Ad Hoc Networks.* Springer, 2007.

[9] Y. Wang, W. Wang, and X.-Y. Li. *Distributed low-cost backbone formation for wireless ad hoc networks. In: Proceedings of the 6th ACM International Symposium on Mobile Ad Hoc Networking and Computing(MOBIHOC'05).* ACM Press, 2005.