数理解析研究所講究録 1375 巻 2004 年 282-288

匿名ネットワーク上のリーダー選出問題の空間計算量

安藤 映 (Ei Ando) 小野 廣隆 (Hirotaka Ono)
定兼 邦彦 (Kunihiko Sadakane) 山下 雅史 (Masafumi Yamashita)

Department of Computer Science and Communication Engineering,
Graduate School of Information Science and Electrical Engineering,
Kyushu University

概要

匿名ネットワークは識別子を持たないネットワークであり，その上でのリーダー選出問題の困難さはプロセッサに与えられる初期大域情報の量に依存する．ネットワークのトポロジーを G, その位数を n とすると，n が与えられないネットワークでは自明な場合を除いてリーダー選出は不可能であるのに対して，n は G とちょうど同じ情報を持つことが知られている．本稿では，n が初期大域情報として与えられるという条件の下で, リーダー選出問題の空間計算量，すなわち問題を解決するためにプロセッサが使用する記憶量が $O(n \log d)$ ビットであることを示す．ここで，d はネットワークの最大次数である．さらに，上記の証明に用いる分散アルゴリズムは任意の n に対して正しく働くのに対し，任意に固定された n に対してのみ働くリーダー選出アルゴリズムでわずか 1 ビットの記憶しか用いないものが存在することを示す．

最後に，固定されていない n に関するリーダー選出問題の空間計算量の下限について議論し，一つの下限 $\Omega(\log n)$ ビットを導出する

1 はじめに

リーダー選出問題とはプロセッサの相互通信により，ある一つのプロセッサを選出する問題である．リーダー選出問題の本質はプロセッサ間の対称性の打破にあり，これには様々な手段が考えられる．たとえば，特別な識別子を持つプロセッサは識別子の相違によって，特別な次数を持つプロセッサは次数の相違によって他と区別される．しかし，識別子が存在せず，トポロジーが正則グラフとなるようなネットワークにおいてもリーダーが選出できる場合があることは知られていた．では，プロセッサが区別できるための条件は何か? 匿名ネットワークは識別子を持たない (あるいは同じ事であるが，同じ識別子を持つ) プロセッサから構成されるネットワークである．匿名ネットワークの研究はこの疑問に答えるために開始された．

山下と亀田はある決定性アルゴリズム A_G

によってリーダー選出が可能な匿名ネットワーク G を特徴付けるとともに，そのようなネットワークの集合 $\Pi = \{G\}$ に含まれるどのネットワーク G 上でも，G の位数 n が初期情報として利用できるならばリーダー選出を正しく行なうことができる分散アルゴリズム A を構成した [2]. あるネットワーク G 上のすべてのプロセッサ u が A を実行すると, 1) $G \notin \Pi$ ならばリーダー選出が不可能であることを認識して停止し, 2) $G \in \Pi$ ならばあるプロセッサ u_0 を選出する．すなわち，u_0 は選出されたことを，それ以外のプロセッサは選出されなかったことを認識して停止する．A はビューの概念に基づいている．ビューはネットワークのあるプロセッサから開始されるすべての道をその道から得られるすべての情報とともに木の形に整理したものである．しかし，ビューの大きさは n に関して指数的であり，したがって，A の空間計算量は n の指数関数となる．

我々はリーダー選出問題の空間計算量がプロセッサ当たり $O(n \log d)$ ビットであることを，具体的に条件を満足する分散アルゴリズムを構成することで示す．ここで，d はネットワークの最大次数である．このアルゴリズムは任意の n に対して正しく働く．さらに，任意に固定された n に対してのみ働くリーダー選出アルゴリズムならばわずか 1 ビットの記憶しか用いないものが存在することを示す．

本稿の構成は以下の通りである．第 2 節でモデルについて説明した後，まず固定された n に対して働く 1 ビット空間計算量アルゴリズムを第 3 節で説明する．その後，第 4 節では任意の n に対して働く $O(n \log d)$ ビット空間計算量アルゴリズムを説明する．最後に第 5 節では任意の n に対して働くアルゴリズムの空間計算量について議論し，一つの下限 $\Omega(\log n)$ ビットを示す．

2 匿名ネットワークとビュー

2.1 匿名ネットワーク

匿名ネットワークを無向連結グラフ $G = (V, E)$ によってモデル化する．頂点集合 $V = \{v_1, v_2, \ldots.., v_n\}$ がプロセッサ集合を表し，辺集合 $E = \{(u, v) | u, v \in V\}$ がプロセッサ間の通信リンク集合を表す．v_i などの名前は記述のためにのみ用いるものであって，アルゴリズムからは一切利用できない．プロセッサ v の次数を $deg(v)$，G の最大次数を d で表す．

プロセッサ v はポート番号 $\{1, 2, \ldots, deg(v)\}$ を通信リンクにつけることによって区別する．通信リンク $\{u, v\}$ について u がつけているポート番号を $\lambda_{(u,v)}$ と書き，$\Lambda = \{\lambda_{(u,v)} | \{u, v\} \in E\}$ をグラフ G のラベル付けとよぶ．匿名ネットワーク N はグラフとラベル付けの対 $N = (G, \Lambda(G))$ である．匿名ネットワークの例を図 1 に示す．

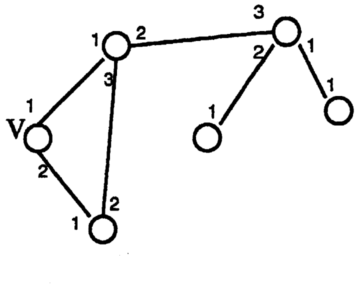

図 1: ネットワークの例

プロセッサでは通常の内部計算の他にプロ

セッサ間通信が可能である．プロセッサ間の通信は通信リンクを介した非同期メッセージパッシングによって実現される．プロセッサ u はある指定した局所記憶に保持されているメッセージ M をポート名によって指定される通信リンク L から送信できる．M は L によって u と接続するプロセッサ v の (通信リンク毎に設置された) 入力キュー Q にある (予想できない) 通信遅延を伴って挿入される．ただし，全てのの通信リンクは FIFO であり，通信リンクの中でのメッセージ順序は保存される．Q は v の局所記憶ではない．v が M にアクセスするには M を v の局所記憶に移す必要がある．L に対する v のポート番号 j によって v は Q にアクセスする．j と局所記憶を指定すると，Q におかれている最初のメッセージが指定された記憶領域に格納される．これが受信事象である．したがって空間計算量は少なくとも利用するメッセージの長さ以上である．なお，Q か空か否かを指定した局所記憶領域に返す述語 $Empty(j)$ が利用できるものと仮定する．

各プロセッサは同じアルゴリズム A で動作する．A が以下の条件を満たすとき，A はリーダー選出アルゴリズムであると言う．

1. リーダー選挙が可能ならば一つプロセッサを選出する．
2. (A にとって) リーダー選出が不可能ならばそのことをプロセッサに報告する．

2.2 ビューについて

ネットワークが与えられているとする．プロセッサ v のビュー $T(v)$ は $v \in V$ から出発し，そこから辺をたどって進むことのできる路を全て集め，無限に続くラベルつきの木の形にしたものである．$T^k(v)$ を v のビューから深さ k より深い部分を取り除いたラベルつきの木とし，深さ k までのビューと呼ぶことにする．図 1 のネットワーク中の v の深さ 3 までのビューを図 2 に例示する．

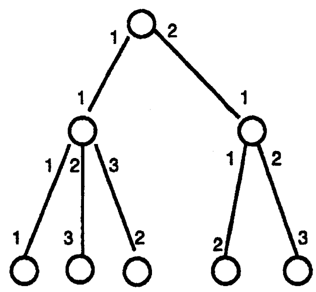

図 2: 深さ 3 までのビューの例

二つのプロセッサ u, v のビューが (ラベルを保存して) 同型であるとき $T(u) = T(v)$ と書く，二つのビューの同型性は次式から深さ $n-1$ までのビューの同型性に一致する [1].

$$T(u) = T(v) \Leftrightarrow T^{n-1}(u) = T^{n-1}(v) \quad (1)$$

$T(u) = T(v)$ ならば最悪の場合に u, v はまったく同じ動作をするので，u と v の対称性は打破できない．これがビューとリーダー選出問題との基本的な関係である．

3 1ビット空間計算量アルゴリズム

本節ではある固定された n に対してリーダー選出を行う1ビット空間計算量のアルゴリズムを構成する.

3.1 1ビット通信に基くリーダー選出アルゴリズム

山下と亀田のリーダー選出アルゴリズムでは, 各プロセッサが深さ $2n-1$ までのビューを構成できれば, プロセッサ内部の計算でリーダー選出問題を解決できた. ここでは, ビュー構成後の内部計算は同様の手法を用いると考えて, ビット通信に基いてビューを構成する方法をだけを説明する.

準備のために, メッセージ長を1ビット数に制約しても, プロセッサに十分な記憶があればビューが構成できることを示す. まず, $(b_1, b_2, b_3) \in \{0,1\}^3$ の3ビットの組を用いてビューを表現する. b_1, b_2, b_3 の組は表1に示すように記号と対応させることにより, これ

記号	b_1	b_2	b_3
(	0	0	0
)	1	0	0
0	0	1	0
1	1	1	0
.	*	*	1

表1: ビット列と記号との対応

によりビットの並びで $\Sigma = \{(,),1,0,.\}$ からなる記号列を表現する.

次にビューを Σ からなる記号列で表現する. 木構造は $\{(,)\}$ によって表現できることが知られている. 一つの節点を一組の () で表現し, その子節点は括弧の間に書くのである. ビューはラベルつきの木であるので根を除く各節点の前にラベルを表す数値を二進数で表記することにする. こうして任意の深さのビューが Σ 上の記号列で表現できる. 二進数の区切りを"." で表現すると, 例えば図2の深さ3までのビューは (1.1(1.10()10.11()11.10())10.1(1.1()10.11())) となる. これを1の1進数で表現し, 0を区切り記号とする1ビットのメッセージ交換を繰り返すことでビューを交換/構成できる.

3.2 1ビット空間計算量アルゴリズムの構成

上記の方式では, メッセージ長は1ビットになったが, ビューを構成するためにはビューの大きさと同じ程度の個数の記憶が必要になる. 本節では, 記憶量を1ビットにまで削減する.

$G_i = (V, E_i)$ をある n 頂点で連結な無向グラフとし, Λ_i を G_i のラベル付けであるとする. そして, 匿名ネットワーク $N_i = (G_i, \Lambda_i)$ $(i = 1, 2, \ldots)$ に関して $\mathcal{N} = \{N_i\}$ とすると $|\mathcal{N}|$ は有限である.

ここで, A を3.1で示した手法にしたがってビューを構成するリーダー選出アルゴリズムとする. ネットワーク N_i 中の v における計算の最初から t 番目のアルゴリズムコードを $com(N_i, t)$ と書くことにする. A は停

止するので列 $com(N_i,1), com(N_i,2), \ldots$ は必ず有限であり，N_i における v の実行履歴 $C_v^i = \{com(N_i,1), com(N_i,2), \ldots\}$ を定義する．アルゴリズムは $i = 1,2,\ldots$ に関して C_v^i を並べたものに対して置換を行ったものである．具体的には主記憶への書き込みを行う部分全てを条件分岐に置き換え，受け取ったメッセージと整合する実行履歴 C_v^j へジャンプする．今，全ての N_i に関する実行履歴をアルゴリズムコードとして並べているので必ずそのような C_v^j は存在する．

プロセッサの計算はまず C_v^1 から開始し，主記憶に書き込む代わりに C_v^1 の実行履歴と受け取ったメッセージが整合するかをチェックし，整合しない場合に別の実行履歴へジャンプし，計算を継続する．

$i = 1,2,\ldots$ に関して C_v^i は全て有限なのでこの計算は必ず終了し，リーダー選出問題を解くことができる．よって，固定された n に対しては 1 ビット空間計算量でリーダー選出問題を解けることが示された．

4 リーダー選出問題の空間計算量

第 3 節では，固定された n に対してのみ正しく働くリーダー選出アルゴリズム A_n の空間計算量を議論した．しかし，多くの場合に議論の対象となるのは，どの n が与えられた時にも正しく働くリーダー選挙アルゴリズムである．もちろん，n が与えられたときに A_n をまず計算し，それを実行するという戦略を取ることは可能である．しかし，そのためには，A_n の計算と保持に必要な記憶量をリーダー選挙アルゴリズムの空間計算量として計上する必要があり，特に第 3 節で示したアルゴリズムはこの目的のために利用するにはアルゴリズム長が大き過ぎる．

本節では任意の n に対して働くプロセッサ当たりの空間計算量 $O(n \log d)$ ビットのリーダー選出アルゴリズムを示す．図 3 に示す構造のメッセージを利用する．$f_1, f_2, \ldots, r_1, r_2, \ldots$ は 0 から d までの整数

f_1	r_1	f_2	r_2	$\ldots$	c	b

図 3: メッセージの構造

を格納する領域で，一つあたり $\log_2(d+1)$ ビットである．c は 0 から $2n-1$ までの整数を格納する領域であり，したがって $\log_2(n-1)$ ビットである．b は 1 ビットを格納する領域である．

直観的には，$f_1, f_2, \ldots$ はビューの下へ進む道順で，$r_1, r_2, \ldots$ はビューの上へ戻る道順である．c はメッセージが最初に送信されたプロセッサからの距離をあらわし，b はメッセージがビューの上へ向かうか下へ向かうかを表すビットである．このメッセージを一つ送信し，それに対するレスポンスを待つことでビューの中の一本の路がどのような形をしているかを見ることができる．

プロセッサ u, v のビュー $T(u), T(v)$ について $T(u) \neq T(v)$ であるとき，$T(u)$ と $T(v)$ に順序をつける．今回の順序付けは以下のようなものを用いる．

まず全ての路に順序をつけ，最初の路から順に路が $T(u), T(v)$ に存在するかどうかを調べる．道順の順序は道順中のポート番号ラベ

ルを d 進数の一桁とみなすと，道順はある整数とみることができ，この整数の大小関係を用いる．$T(u) \neq T(v)$ であるならば，ある道順 l_k が存在して，

$$l_k \notin T(u) \wedge l_k \in T(v)$$
$$\text{または}\ \ l_k \in T(u) \wedge l_k \notin T(v).$$

このような l_k のうち，道順の順序で最初のものを l_1 とすると，l_1 を用いて $T(u), T(v)$ の間の順序 $<$ を以下のように定義する．

$$l_1 \in T(u) \wedge l_1 \notin T(v) \Rightarrow T(u) < T(v)$$

この順序では明らかに次式が成立する．

$$\neg(T(u) < T(v)) \ \wedge\ \neg(T(v) < T(u)) \Rightarrow T(u) = T(v)$$

図 3 に示した構造のメッセージで一本の路の形を見ることができるため，このメッセージを二つ同時に送信することで二つのビューの中でのある道順 l の存在を調べることができ，したがって二つのビューの順序を知ることができる．

そして，深さ $n/2$ までのビューの中の全てのの二つの節点の組合せに関してビューの比較を行うことで異なる構造のビューの数を数えることができる．

以下の定理 1,2 [2] によって互いに同型ではないビューの個数が $(n+1)/2$ より少なければリーダー選出が不可能なネットワークであり，逆に $(n+1)/2$ 個より多ければリーダー選出が可能である．

定理 1 [2] 互いに同型ではないビューの個数はある自然数 s によって n/s と表される．

定理 2 [2] 互いに同型ではないビューの個数が n のとき，かつそのときに限りリーダー選出可能である．

4.1 空間計算量の下限

本節では，リーダー選出問題の空間計算量の下限にを示す．

定義 二つの事象 e, f に関して，因果関係 $\prec$ を以下のように定義する．

1. e と f は同一プロセッサ上の別々の事象であり，e が f の前に発生しているならば $e \prec f$.
2. メッセージを送信事象 e と対応する受信事象 f に関して $e \prec f$.
3. e, f, g をある事象とすると，$e \prec g \prec f \Rightarrow e \prec f$.

$e \prec f$ が成立するとき，e と f の間には因果関係があるという．

補題 1 アルゴリズム A が以下の条件を満足すると仮定する:

> A がネットワーク N をリーダー選出不可能と結論したときにはその結論は常に正しい．

このとき，任意の二つのプロセッサ u と v について，以下の命題が成立する:

> u のある事象 e と v のある事象 f の間には因果関係がある．

証明 (概略) A をリーダー選出不可能なネットワーク $N = (G, \Lambda), G = (V, E)$ に対してだけ「リーダー選出不可能」の結論を出して停止するアルゴリズムとする．$u, v \in V$ に関して v が u からのどのメッセージとも因果関係を持たずに計算を終了すると仮定する．

ある 3-辺彩色可能な 3-正則グラフ G について，次の条件を満たす G のラベル付けの組 (Λ_1, Λ_2) が存在する．Λ_1 と Λ_2 は $u \in V$ のラベル付けだけを変更したものであり，$N_1 = (G, \Lambda_1)$ はリーダー選出不可能で，一方 $N_2 = (G, \Lambda_2)$ はリーダー選出可能である．このようなラベル付けは全ての辺の両端のラベルが同一であるようなものを N_1 と考えると容易に存在を示すことができる．仮定より v は N_1 と N_2 を区別することができず，どちらの場合にも同じ結果を出力する．よって矛盾．■

この補題を用いて空間計算量の下限 $\Omega(\log n)$ を示すことができる．

定理 3 匿名ネットワークリーダー選出問題の空間計算量は $\Omega(\log n)$ である．

証明 (概略)　補題 1 より，v は全てのプロセッサ $u \in V$ に関して少なくとも一つの u の送信事象と因果関係が無くてはならない．したがって，一つのプロセッサが停止するまでにメッセージ数 $n-1$ が必要である．

背理法で証明するために，各プロセッサは $n-2$ 個の状態しか取れないと仮定する．すると，全てのプロセッサが同期的に動作するような場合にはネットワークの大域コンフィグレーションは $n-2$ しか存在しないことになる．このため，ある大域コンフィグレーション C が存在して $n-1$ 個のメッセージを交換する間に 2 回現れることになり，計算を停止することができない．

このことから，$\Omega(\log n)$ ビットの空間計算量が得られる．■

5 結論

本稿では匿名ネットワーク上でのリーダー選出問題に対して，空間計算量の観点から考察を行った．まず，固定された n についてのみリーダー選出を行うアルゴリズムの中には，1 ビットの記憶しか利用しないものが存在することを示した．また，一つのアルゴリズムで一般の n に対応する場合には，$O(n \log d)$ ビットの空間計算量のアルゴリズムを示した．空間計算量の下限については $\Omega(\log n)$ ビットを示したが，まだ上下限の間に開きがあるため，より厳密な下限の存在が予想される．

参考文献

[1] N.Norris, "Universal covers of graphs: isomorphism to depth $n-1$ implies isomorphism to all depths", Discrete Applied Mathematics, 56, 1995, 61-74.

[2] Masafumi Yamashita and Tsunehiko Kameda , "Computing on Anonymous Networks: Part 1 - Characterizing the Solvable Cases" , IEEE trans. on parallel and distributed systems, Vol.7 , No.1, January 1996.