拡張独立点集合問題を解く自己安定分散アルゴリズム

2Q-05

川本 幸司　　角川 裕次　　阿江 忠

広島大学工学部

1　はじめに

分散システムとはプロセスと通信リンクの集合から構成されるシステムである．通常の分散アルゴリズムでは，ネットワークの初期状況があらかじめ決められている．これに対し，これに何も仮定をしないものを自己安定分散アルゴリズムという．これ以降は自己安定分散アルゴリズムを、自己安定アルゴリズムと呼ぶ．

分散システムでは，問題を解決するために特別なプロセスを選出することが必要になる場合があり，ある一つのプロセスを選出する問題をリーダー選出問題という．リーダー選出問題は基本的な分散問題であるので，これまでによく研究されている．

非同期，あるいは同期メッセージ伝達モデルで故障を仮定しない場合での様々な結果は，例えば文献 [1] を参照されたい．また，一般のネットワーク上でのリーダー選出問題の可解性については，文献を [2] を参照されたい．

自己安定リーダー選出問題に対するこれまでの結果は以下の通りである．文献 [3] において Huang はプロセス数が素数である双方向リングネットワークでの決定性アルゴリズムを提案している．(プロセス数が合成数の場合は，決定性のアルゴリズムは存在しない．) 文献 [4] では，Autonioiu と Srimani らによって木ネットワークでのアルゴリズムが提案されている．

グラフ理論での基本的な問題の 1 つとして，独立点集合問題がある．独立点であるノード (或はプロセス) は隣接プロセスの情報によって決定できるのに対して，リーダー選出問題では大局的な情報がなければリーダーを選出できない．そこで本稿では，これら 2 つの問題を特別な場合として含む拡張独立点集合問題を提案し，その自己安定解を調べることを目的とする．

2　モデル

分散システムは n 個のプロセスの集合 $C = \{p_i \mid i = 1, 2, 3, ..., n\}$ と m 本の双方向の通信リンクの集合 $L = \{..., (p_i, p_j), ...\}$ から構成される．$(p_i, p_j) \in L$ のときは，p_i, p_j 間にリンクが存在する．この時，p_i，p_j は互いに隣接プロセスと呼ぶ．N_i を，p_i の隣接プロセスの集合とする．各プロセスは局所状態を持ち，どのプロセスも隣接プロセスの局所状態が参照できる．

各プロセスに与えるアルゴリズムは，以下のようなガード付きコマンドの形で記述する．

$$
\begin{array}{l}
*[\\
\quad guard_1 \ \rightarrow \ command_1 \\
\quad guard_2 \ \rightarrow \ command_2 \\
\quad ... \\
\quad guard_N \ \rightarrow \ command_N \\
]
\end{array}
$$

各ガードは，p_i 及び N_i の状態に関する述語である．各コマンドは，p_i と N_i の状態より，次の p_i の状態を決定する文である．プロセス p_i が真であるガードを持つとき，プロセス p_i は特権をもつという．

すべての隣接プロセスの状態を読み込み，その情報と自身の状態のみをもとに自分の状態を更新する一連の動作を，プロセスの 1 原子動作 (atomic step) とする．

動作を行うプロセスを選択するスケジューラーとして C デーモン（*Central daemon*）と D デーモン（*Distributed daemon*）とがある．

デーモンは各ガードをチェックし，特権をもつプロセスの中から，実行すべきプロセスを選ぶ．選ばれたプロセスは真であるガードに対応するコマンドを 1 つ選んで実行する．この時，選ばれたプロセスが複数あるときは，並列実行される．デーモンはこの一連の動作を繰り返す．

C デーモンは特権を持つプロセスを一つ選ぶ．これに対し D デーモンは特権を持つプロセスの内一つ以上の任意の数のプロセスを選ぶ．

定義 1.(状況)：各 $i(1 \leq i \leq n)$ に対して，プロセス p_i が取り得る状態の集合を Q_i とする．システム S の状況 (*configuration*) とは，全プロセスの状態の組 $\gamma \in Q_1 \times Q_2 \times \cdots \times Q_n$ である．全ての状況の集合を $\Gamma = Q_1 \times Q_2 \times \cdots \times Q_n$ と表す．□

定義 2.(状況の遷移)：任意の状況 $\gamma \in \Gamma$ に対して，1 ステップの実行により状況が γ' となるとき，$\gamma \rightarrow \gamma'$ と書く．□

定義 3.(自己安定)：システム S のアルゴリズム A が，$\Lambda \subseteq \Gamma$ に関して自己安定であるとは，以下が成立するときをいう．

1. 閉包性．任意の $\lambda \in \Lambda$ に対して，$\lambda \rightarrow \gamma$ ならば $\gamma \in \Lambda$.
2. 収束性．任意の $\gamma_0 \in \Gamma$ と任意の実行 $\gamma_0 \rightarrow \gamma_1 \rightarrow \cdots$ に対して，ある i が存在して $\gamma_i \in \Lambda$．Λ を正当な状況の集合という．□

定義 4.(拡張独立点集合)：各プロセス p_i の状態集合を Q_i とし，関数 $f_i : Q_i \rightarrow \{0, 1\}$ を定める．ある d に対して，アルゴリズム A が自己安定拡張独立点集合問題を解くとは，正当な状況の集合 Λ に対して以下が成立するとき，及びそのときのみを言う：

- 各 $\lambda = (q_1, q_2, \cdots, q_n) \in \Lambda$ に対して $I_\lambda = \{p_i | f(q_i) = 1\}$ とするとき，任意の $p_i, p_j \in I_\lambda$ のネットワーク距離は d より大きい．
- 各 $\lambda \in \Lambda$ に対して，$|I_\lambda| > 0$ である．
- $\Lambda \neq \phi$ □

本稿では，一般性を失うことなく，各プロセス p_i は同一の状態集合 Q を持つものとする．独立点集合問題に対して極大あるいは最大独立点集合問題が定義されるのと同様に，拡張独立点集合問題においても極大及び最大拡張独立点集合問題を定義できる．

The Expanded Independent Set Problem.
and the Self-stabilizing Distributed Algorithm
Kouji Kawamoto, Hirotsugu Kakugawa, Tadashi Ae
Faculty of Engineering, Hiroshima University

3 プロセス ID を持たない場合の結果

定理 1:任意のネットワーク形状に対して，C デーモンの下で極大独立点集合問題を解くアルゴリズムが存在する．($d = 1$ の場合) □

定理 2: 任意の d に対して、ネットワークの形状が正則グラフの場合，D デーモンの下で問題を解くアルゴリズムは存在しない． □

定理 3: 各 $d > 1$ に対して C デーモンの下で動作する時，プロセス数 $n = dx(x \geq 1)$ のリングネットワークでは，拡張独立点集合問題を解くアルゴリズムは存在しない． □

定理 2，3 より分かるように，ネットワーク形状に対称性があり，$d > 1$ の場合にはアルゴリズムが存在しない．次の章ではプロセスに一意な識別子がある場合を考える．

4 プロセス ID を持つ場合の結果

4.1 最大拡張独立点集合問題を解くアルゴリズム

D デーモンの下で最大拡張独立点集合問題を解くアルゴリズムの概要は以下の通りである．

1. 既存のリーダー選出問題を解くアルゴリズムを用いてリーダーを選ぶ．
2. (1) で選出されたプロセスを根とする生成木を生成する．(例えば，Arora らによる文献 [5] のアルゴリズムを用いる．
3. ネットワーク上の全てのプロセスに一意な識別子を割り当てる．(例えば，文献 [6] の Datta らによるアルゴリズムを用いる．
4. 生成木の各葉プロセスから親プロセスに向かってネットワーク形状 G を送る．
5. 根プロセスは全プロセスに対して G を放送する．
6. 各プロセスは，G に対して最大拡張独立点集合問題を解く．その結果を I とする．(各プロセスが持つ I は同一であることに注意)
7. 各プロセス p_i は，$p_i \in I$ であるならば独立点となる． □

この手法では各プロセスは大域情報を得る必要があるのと，(6) での計算は NP 完全問題の計算である．以降では局所的な情報とより少ない計算複雑度で解くアルゴリズムについて考える．

4.2 D デーモンの下でプロセス ID を持つ場合

各プロセスが一意な識別子を持つ場合において拡張独立点集合問題を解くアルゴリズムを提案する．アルゴリズムの方針は以下の通りである．

各プロセスは，自分の識別子が距離 d 以内のプロセス群の中で最小の時及びその時のみに限って独立点となる．提案するアルゴリズムでは，大局情報の収集はせずに，各プロセスの識別子を距離 d の範囲内にしか伝えないようにする．

アルゴリズムでは，距離 $k(1 \leq k \leq d)$ 以内のプロセスの最小の識別子を保持するために，配列変数 $m_i[1...d]$ を用いる．id_i をプロセス p_i の識別子とし，便宜上 $m_i[0] = id_i$ とする．また，変数 $v_i \in \{0, 1\}$ は，プロセス p_i が独立点である時及びその時のみに限って 1 である変数である．

このアルゴリズムの動きを直観的に言えば，まず各プロセス p_i の $m_i[1]$ が収束し，それに基づいて各プロセス p_i の $m_i[2]$ が収束する．順次 $k = 3, 4, ..., d-1$ に対して $m_i[k]$ が収束し，最終的には $m_i[d]$ が収束する．

定理 4:提案するアルゴリズムは，拡張独立点集合問題を解く． □

5 準均一な識別子を持つ場合

プロセス群が準均一な識別子を持つとは以下の場合をいう．

- 根 (root) プロセスと呼ばれる特別なプロセスがある．
- 根プロセスの識別子は他のプロセスと異なるが，その他のプロセスの識別子はいずれも同じである．

定理 5: 任意の d，任意のネットワーク，任意のデーモンに対する拡張独立点集合問題の可解性は，識別子が一意である場合と準均一である場合ともに同じである． □

6 おわりに

拡張独立点拡張問題を解く自己安定アルゴリズムの可解性についてのいくつかの結果を示した．各プロセスが固有の識別子さえもっているならば，D デーモンを仮定した場合でも問題を解くことが可能であることがわかった．しかし，逆に各プロセスが一意な識別子がなくネットワークが対称な場合，問題を解くアルゴリズムは存在しない．文献 [2] では，リーダー選出問題の可解性をビュー (*view*) を用いて議論している．拡張独立点集合問題の可解性も同様な手法で議論できるものと思われ，今後の課題とする．

謝辞 本研究の一部は，文部省科学研究費補助金 (課題番号 11780229) の援助を受けている．

参考文献

[1] 亀田恒彦, 山下雅史：分散アルゴリズム, 近代科学社 (1994).

[2] Yamashita, M., Kameda, T.: Leader Election Problem on Networks in which Processor Identity Number are Not Distinct, *IEEE Transactions on Parallel And Distributed Systems*, Vol. 10, No. 9, pp. 878–887 (1999).

[3] Huang, S.-T.: Leader Election in Uniform Rings, *ACM Trans. Program. Lang. Ayst.*, Vol. 15, No. 3, pp. 563–573 (1993).

[4] Gheorgh Antonioiu, P. K. S.: Self-Stabilizing Leader Election Algorithm for Tree Graphs, *Journal of Parallel And Distributed Computing*, Vol. 34, pp. 227–232 (1996).

[5] Arora A, M., and Gounda: Distributed Reset, *IEEE Transactions on Computers*, No. 43, pp. 1026–1038 (1994).

[6] K.Datta, A., Gurumurthy, S., Petit, F., Villain, V.: Self-Stabilizing Network Orientation Algorithms in Arbitrary Rooted Networks, *ICDCS* (2000).