

センサネットワークを対象とした分散アルゴリズムの自律適応性と安定性に関する研究

研究代表者 山内 由紀子 九州大学大学院 システム情報科学研究院 助教

1 はじめに

小型の低能力な無線センサ端末多数で構成されるセンサネットワークは、環境観測等多くの分野で利用が広がっている。各無線センサ端末は目的に合わせたセンサ(温度センサ、照度センサ、加速度センサ等)を備えており、他の端末と無線通信を介して協調しながら観測データの収集を行う。自律的に動作する計算機群が相互に通信を行いながら協調動作するシステムを分散システムと呼ぶ。分散システムを効率よく運用するためのアルゴリズム(分散アルゴリズム)は従来から多くの研究が行われている。特に、分散システムを人間の介入なく運用することを可能とするような、自律適応性、自己組織化能力といった性質を持つ分散アルゴリズムの研究が多く行われている。センサネットワークは分散システムの一つととらえることができるため、既存の分散アルゴリズム設計手法を適用できる可能性があるとともに、センサネットワーク特有の問題を反映した新たな分散アルゴリズムの設計を必要とする可能性がある。本研究では、センサネットワークの持つ動的な環境変化に自律的に適応する分散アルゴリズムの設計を目指す。

センサネットワークでは、センサ端末の故障、電池切れによる停止、無線通信の使用状況の変化、観測データの増減など、システムの状況が時々刻々と変化する。このような環境下の分散システムには、変化に追従して自身の振る舞いを変化させる自律適応性と、変化の下でも安定した振る舞いを行う安定性の両方の性質を持つことが必要である。しかし、この2つの性質を同時に備えた分散システムの理論的な研究は十分に行われているとは言い難い。

優れた自律適応性を持つ分散アルゴリズムの一種として、自己安定アルゴリズムがよく研究されている[4, 5]。しかし、従来の自己安定アルゴリズムでは安定性が十分に考慮されていない。本研究では、自己安定アルゴリズムに安定性を付加することにより、安定性と自律適応性を同時に備えた分散システムの設計手法を確立することを目指し、いくつかの成果を得た。また、センサネットワークで発生する大量の観測データの収集、処理手法についてもいくつかの成果を得た。以下に本研究での成果をまとめる。

2 自己安定アルゴリズムの安定性

2-1 研究背景

どのような初期状況から実行を開始しても、やがて目的の動作を行う状況に復帰する分散システムを自己安定であると言う[4, 5]。自己安定システムは任意の有限個の一時故障(メモリのソフトウェア等)に対して、最後の故障発生後にシステムが自律的に復帰することを保証するため、優れた自律適応性、故障耐性をもつ。自己安定な分散システムを実現する分散アルゴリズムを自己安定アルゴリズムと呼び、リーダー選挙問題、全域木作成問題、極大独立点集合問題、トークン巡回による相互排除といった、計算機ネットワークを運用する上での基礎的な機能を実現する自己安定アルゴリズムがよく研究されている[4, 5]。

一般に自己安定アルゴリズムでは、各計算機が絶えず他の計算機と情報を交換しながら、自身の計算結果(を格納する変数の内容)を更新することで、システム全体が状況の変化に追従することを実現している。しかし、自己安定アルゴリズムは明示的な計算終了判定を行わず、アルゴリズムの実行途中に各計算機がもつ計算の途中経過をシステム外部のユーザやソフトウェアに観測されることを許すため、実行途中に信頼性のある機能を保証できない。さらに、計算の途中経過が更新される度に、外部のユーザ、ソフトウェアが影響を受け、ネットワーク全体に大きな影響を与える可能性がある。

では、実行途中の各計算機での更新回数を削減することは可能であろうか。たとえば、ネットワーク中に全域木を作成する全域木作成問題では、各計算機が正しい計算結果を得るためには、本質的にある種のネットワーク全体の状況を知る必要がある。ネットワーク中で直接接続していない計算機どうしは、他の計算機が情報を中継することにより互いの状態を得るため、この情報伝搬の間、各計算機で不要な更新が発生する

可能性がある。本研究では、まず、各計算機がネットワークのどの程度の範囲の情報を得られれば、更新回数を最小化できるかを検討した。その結果、分散環境で扱われる問題の多くは、各計算機がネットワーク全体の情報を必要とすることがわかった。この結果をもとに、本研究では更新回数を最小化する自己安定アルゴリズムの実現可能性を議論する。

2-2 分散システムと自己安定アルゴリズム

本節では、次節以降、本報告書内で使用する分散システムの諸定義を示す。

分散システムは計算機間の相互接続関係を表す無向グラフ $G=(V, E)$ で表わされる。1つの頂点が1台の計算機を表し、相異なる計算機 p, q が互いに直接通信可能であるとき、辺 $(p, q) \in E$ であるとする。ただし、 G は一般のグラフであるとする。2つの相異なる計算機 $p, q \in V$ に対し、 G 上での p - q 間の最短パスの長さをこの計算機間の距離と言う。各計算機は状態機械であり、局所変数の集合と局所アルゴリズムを持つ。計算機の状態は局所変数の値で表し、局所アルゴリズムは状態機械の状態遷移関数である。すべての計算機の局所アルゴリズムの集合を分散アルゴリズムと呼ぶ。各計算機の局所変数は、入力変数（アルゴリズムへの入力を格納する変数）、出力変数（アルゴリズムの計算結果を格納する変数）、内部変数（その他の変数）から成る。局所アルゴリズムは入力変数と内部変数を用いて計算を進め、出力変数に計算結果を格納する。すべての計算機の出力変数の値が分散アルゴリズムの出力である。

分散アルゴリズムの計算の進行を把握するため、システムの状態と実行という概念を導入する。分散システム中の全計算機の状態の組を状況と呼び、システムの状態の系列でアルゴリズムの実行を表現する。実行 $\mathcal{E}=\mathbf{c}_0, \mathbf{c}_1, \dots$ が与えられた時、各状況遷移 $\mathbf{c}_i \rightarrow \mathbf{c}_{i+1}$ ($i > 0$) は以下のように決まる：スケジューラが計算機の任意の部分集合を選択し、選択された計算機が各々の局所アルゴリズムを実行することで状況 \mathbf{c}_i から次状況 \mathbf{c}_{i+1} へとシステムの状態が遷移する。スケジューラは通信遅延や計算機の速度の速さの違いといった分散システムの非同期性を反映するための概念である。本研究では弱公平なスケジューラ（局所アルゴリズムを実行したい状態で待機し続けている計算機はいずれスケジューラに選択される）を想定する。

分散システムにおける問題（タスク）は目的とする実行や局所変数への条件として与えられる。たとえば、全域木作成問題[6]は、各計算機が隣接する計算機のうち1つを指すポインタを出力変数として持ち、すべての計算機の出力変数が G 上に根付き内向木を構成することと定義される。

自己安定アルゴリズムは目的とする問題に対し、どのような初期状況から実行を開始しても、有限時間以内に正当な状況に到達すること（収束性）を保証する。任意初期状況から始まるどのような実行も問題の条件を満たす時、この状況を正当な状況と呼ぶ。自己安定アルゴリズムは任意の初期状況から実行を開始しても、やがてシステムが自律的に目的の性質を満たすため、優れた自律適応性を持つ（図1）。また、実行開始時にシステム状況の初期化を必要としない点も、集中型のアルゴリズムとの大きな相違である。

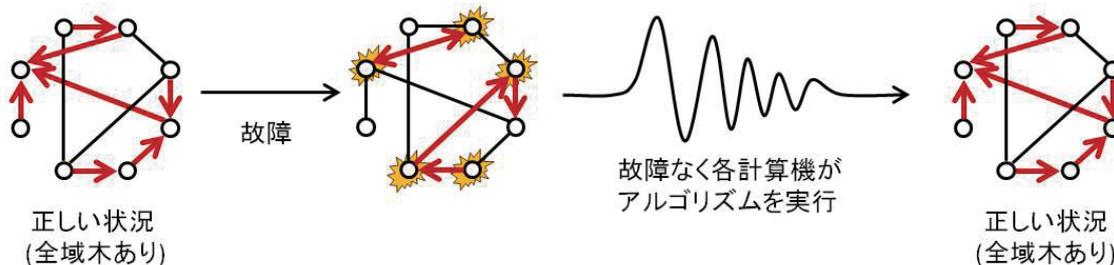


図1：自己安定アルゴリズムの一例（全域木作成問題）

3-2 研究成果

本研究では自己安定アルゴリズムの収束途中における各計算機の状態変更回数に着目した、以下の2つの性質それぞれを持つ自己安定アルゴリズムの実現可能性について示した。

- 単調収束性：どのような実行においても、各計算機は状態を1度でも更新すれば、正当な状況での状態となる。
- 出力単調収束性：どのような実行においても、各計算機は出力変数を1度でも更新すれば、出力変数の値は正当な状況での値となる。

2つの概念の違いは、単調収束性が計算機の状態（つまり、内部変数、出力変数すべて）の更新を対象としているのに対し、出力単調収束性は出力変数のみの更新を対象としている点である。単調収束性では、計算機は自身の状態（入力変数、内部変数、出力変数）と隣接する計算機の状態から、正当な状況での「正しい」状態を計算しなければならない。一方、出力単調収束は、内部変数を用いて計算機どうしで十分に情報交換を行った後に、出力変数を更新する余地がある。出力変数をただ1度しか更新しないという枠組みは分散システムにおける合意問題等でも扱われており、一般的な設定であると言える。

単調収束性は通信やメモリの下界を示すために提案した制約であるのに対し、出力単調収束性は自己安定アルゴリズムとしての実現に興味を置いた制約である。

以下に本研究の結果の概要を示す。

1. 単調収束性の実現可能性：局所的な問題以外については単調収束自己安定アルゴリズムは実現不可能であることを示した。局所的な問題とは、各計算機が隣接する計算機の状態からのみから正当な状況での自身の状態を計算できる問題である。このような問題としては、頂点彩色問題（ G の各頂点にどの隣接頂点とも異なる色を塗る問題）が挙げられる。分散アルゴリズムの多くは、計算機が相互に情報交換を行い、システム全体として何らかの強調動作を行うことを要求するため、このような問題のクラスは非常に小さい。
2. 出力単調収束性の実現可能性：全域木問題、極大独立点集合問題、極大マッチング問題について出力単調収束性を持つ自己安定アルゴリズムは実現不可能であることを示した。本質的な難しさは、任意の初期状況において局所的な情報のみから正しい出力変数の値を得られないことにある。困難さの原理は以下の通りである：出力単調収束性を持つ自己安定アルゴリズムが存在すると仮定すれば、そのアルゴリズムはどのような局所的な状況（隣接する計算機の集合や、それら計算機の状態）であれば計算機が出力変数を更新するかを規定している。ある分散システム G で計算機 p が出力変数を更新する局所的な状況が存在すれば、他の分散システム G' においても同じ局所的な状況で p は出力変数を G と同様の値に更新してしまう。しかし、上記に挙げた3つの問題については、局所的な状況が同一であっても正当な状況では p が異なる出力変数をとるような G, G' が存在することが示せる。たとえば、全域木作成問題においては、 G と G' で全域木の根となる計算機の場所が異なる場合が挙げられる。
3. 各計算機の出力変数の更新回数が定数である自己安定アルゴリズム：極大独立点集合問題、極大マッチング問題について、任意の実行における各計算機の出力変数の更新回数が高々2回であるような自己安定アルゴリズムをそれぞれ作成し、その正当性を証明した。出力変数の更新回数を削減した一方で、ネットワーク全体で情報収集を行うため、既存の自己安定アルゴリズムより実行時間は長い。

3 センサネットワークにおけるデータ収集とストリームデータ処理

センサネットワークを敷設する目的は、対象とする地理領域の観測である。各センサ端末が観測した観測データをシンクと呼ばれる指定されたセンサ端末に収集するデータ収集機能は、センサネットワークのもっとも基本的な機能である。本節では、観測データの収集、また収集したデータの処理手法についての成果の概要を示す。

3-1 研究背景

センサネットワークにおける従来のデータ収集手法の多くは、シンクを根とする根付き内向木を作成し(図2)、内向木の葉から根へと観測データをストア・アンド・フォワードで収集する[1]。しかし、このような手法では、シンク周辺のセンサ端末が多く観測データの中継を行うため、無線通信により電池を消耗し、早期に停止する可能性がある。停止するセンサ端末が発生した場合に、観測領域の一部から観測データを収集できなくなる可能性があるため、センサネットワークの長寿命化を目指す負荷分散手法がよく研究されている[3]。

本研究では、データ収集に用いる経路集合（データ収集トポロジ）をセンサ端末の電池残量、停止や追加に応じて変更する手法を提案する。また、データ収集トポロジにDAGを用いることにより、電池消耗によるセンサ端末の停止があっても、データ収集トポロジが非連結になることを回避する。データ収集においては、中継端末の電池残量を考慮することにより負荷分散を行い、観測データの転送を多重化することでセンサ端末の停止があってもデータが消失なくシンクに配送されることを保証する。

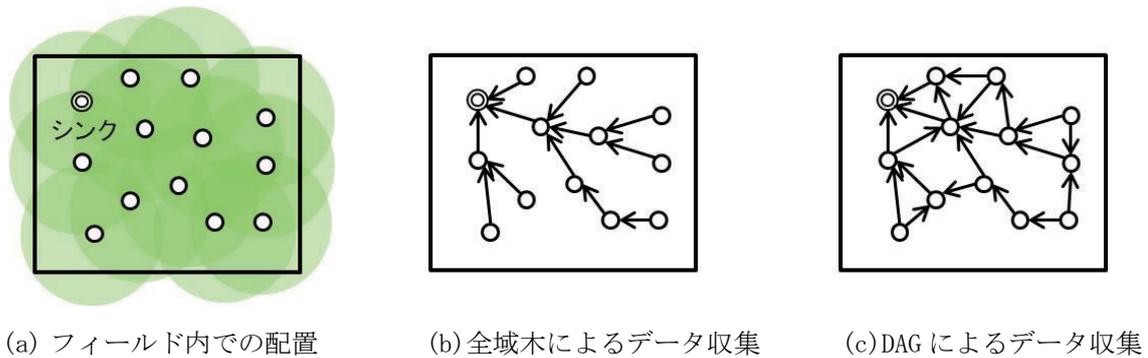


図 2：無線センサネットワーク

無線センサネットワークにおいて、各々のセンサ端末が観測し、シンクに配送される観測データは1つのデータストリームと見なすことができる。大規模なデータストリームに対する効率のよいストリーム処理技術が近年注目を集めており、本研究ではその中でも基本的な問題のひとつである頻出アイテム検知問題に着目する。

頻出アイテム検知は、与えられたストリームデータ中に一定割合以上出現するデータを抽出するものである。単一ストリーム上の頻出アイテム検知問題について、これまでさまざまな研究がなされてきたが[2, 7, 8], いずれもストリーム長 N に対して、 $O(\log N)$ ビットを要する。本研究では、 $O(\log \log N)$ ビットのメモリで、高確率ですべての頻出アイテムを出力する乱択アルゴリズムを示す。提案アルゴリズムは指数近似による頻出アイテムの数え上げのアイデアをもとにしている。使用するメモリサイズより、既存の頻出アイテム検知アルゴリズムと比較してメモリオーバーフローに対してロバストであると言える。

3-2 研究成果：センサネットワークにおける自己安定データ収集手法

(1) センサネットワーク

センサネットワークをグラフ $G=(V, E)$ で表わされる分散システムとする。頂点集合 V はセンサ端末の集合に、 E は無線通信により決まるセンサ端末間の隣接関係を表す。 V 中には、シンクと呼ばれる特別なセンサ端末がただ1つだけ含まれているとする。各センサ端末は電池駆動であり、各時点において自身の電池残量を知ることができる。各センサ端末はデータの送受信、定期的なセンシングを行う。

センシング対象領域内にはセンシングポイントと呼ばれる点集合が指定されているとする。すべてのセンシングポイントがセンサ端末のセンシング半径に含まれる時、センシング対象領域は被覆されていると言う。すべてのセンシングポイントが k 個以上の起動状態のセンサ端末のセンシング半径内にある時、そのような最大の k に対して、センシング対象領域が k 重被覆されていると言う。各センサ端末はセンシング半径内のデータを観測でき、自身が観測できるセンシングポイントを知っているとする。また、センシング半径はセンサ端末の通信距離よりも十分小さく、あるセンシングポイントを観測できるセンサ端末は互いに隣接しているとする。

(2) 提案手法

提案手法は、シンク端末をシンクとする DAG を構成する自己安定アルゴリズムと、この DAG 上でのデータ収集アルゴリズムから成る。

DAG 構成における各センサ端末の処理は以下の4つの手順から成る：

1. シンクからのホップ数計算：幅優先探索を用いた既存の自己安定全域木作成アルゴリズムと同様の手順により、各センサ端末がシンクからの距離(ホップ数)を得る。
2. 各センサ端末の転送先の選択：1で得られたホップ数をもとに、各センサ端末は自身よりシンクに近いセンサ端末のうち1つを転送先として選ぶ。
3. 各センシングポイントの2重被覆の保証：各センシングポイントを2重被覆するためのセンサ端末を2つ選択する。提案手法では、電池残量の多いセンサ端末を選択する。つまり、各計センサ端末はセンシング領域内の各センシングポイントについて、電池残量について上位2位のセンサ端末をそのセンシングポイントの観測者と見なす。各センサ端末には自身のセンシング領域内のセンシングポイントが与えられており、また、あるセンシングポイントをセンシング領域にもつセンサ端末は互に通

信可能であると仮定しているため、各センシングポイントについてちょうど2個のセンサ端末が観測を行うこととなる。

4. 各センサ端末が転送先、迂回先を複数持つ DAG の作成：各センサ端末は隣接するセンサ端末のうち、シンクからの距離が自身と同じものを選択し、迂回先とする。

各センサ端末が取得した観測データは、上記の自己安定アルゴリズムによって作成された DAG に沿って中継され、シンクに配送される。配送途中のデータの消失に備えるため、観測を行うセンサ端末は現在の観測データと直前の観測データを1つのデータとして転送する。1つの観測データにつき、2個のコピーが転送されることになるので、中継途経路上のセンサ端末が1つ停止してもデータが失われることはない。

データ収集トポロジ、データ収集ともに複数のセンサ端末が停止や故障する場合には正当性を保証できない。そのような場合には、アルゴリズムの自己安定性により上記の機能が自律的に回復されることのみが保証される。

3-3 研究成果：ストリームデータからの頻出アイテム検知手法

(1) 頻出アイテム検知問題

Σ をアイテムの有限集合とする。 Σ の要素から成る入力ストリームを系列 $\mathbf{x} = (x_1, x_2, \dots, x_N)$ とした時、指定された閾値 $\theta \in (0, 1]$ に対して、 \mathbf{x} 中に θN 回以上出現するすべてのアイテムを出力せよ。

(2) 提案手法

ストリームデータ処理では、入力ストリームのサイズが予め分からない場合にも正しく動作することが必要である。つまり、どのようなサイズの入力ストリームに対しても、メモリアオーバーフローが生じず、高速に終了することが期待される。つまり、ストリーム全体を一度メモリ上に保存し、複数回走査するのではなく、入力ストリームの各要素に対して、オンラインに処理を行うことが必要である。

提案手法は入力ストリーム中のアイテムの一樣乱択により、頻出アイテムの検知を実現する。ストリーム長 N が既知であれば、適切に選んだ c を用い、一定の割合 c/N で乱択を行えば、所望の精度を保証する一樣乱択は容易である。しかし、ストリームデータでは一般的に事前にその長さ N を知ることは不可能である。

簡単のために、1つのアイテム a から成るストリームデータを想定する。アイテム a が頻出であるかどうか（ストリーム中に θ 割以上出現するか）はストリームの最後の要素が入力されるまで判定できない。ここでストリームの長さを数える問題を考える。

提案手法では既に読み込んだストリーム長に合わせて、乱択確率を減少する。つまり、 K 個のサンプルを保持できる領域を使用できる時、その領域を使用しきるまでは確率 $1/2$ で、その後は $1/2^2, 1/2^3, 1/2^4, \dots$ と乱択確率を減少させていく。乱択確率を減少する際、既にメモリ内に保持しているアイテムに対しては、確率 $1/2$ で再度乱択を行えば、常に一樣乱択したサンプルを保持できる。このような手順で、(現在保持しているサンプル数) \times (現在の乱択確率)⁻¹ がストリーム長の近似値となるような数え上げを行える。また、入力ストリーム中での頻出アイテムは高確率でサンプル中でも θ 割以上出現することが保証できる。

数え上げのために提案手法が必要とするメモリサイズは高確率で $\log \log N$ ビットである。アルゴリズム終了時に乱択確率は高確率で $2^{-\log N}$ であることが保証できる。この確率をそのまま保持するには $\log N$ ビットが必要である。しかし、確率 $1/2$ のコインフリップを $\log N$ 回行えば、 $2^{-\log N}$ を実現することができ、 $\log N$ は $\log \log N$ ビットで表現することができる。 K については、 θ やアルゴリズムの終了確率には依存するものの、 N に無関係に選ぶことができる。

提案する頻出アイテム検知手法の性能は以下の通りである。任意の $\theta \in (0, 1), \gamma \in (0, 1), \delta \in (0, 1)$ に対して、 $K=2^b, b = \lg((\theta\gamma/2)^2 \ln(3((1-\gamma/2)\theta\gamma^{-1}))) + 3$ とした時、任意のストリームに対して以下の出力を行い提案アルゴリズムが終了する確率は $1-\delta$ である：(i) θN 回以上出現するアイテムをすべて含み、(ii) 出現回数が $(1-\gamma)\theta N$ 未満のアイテムを含まない。アルゴリズム全体で使用するメモリは $O((\log^2 \theta^{-1}) / \theta^2 + \log \log N)$ ビットである。計算時間については、提案手法は各入力アイテムに対して平均的に定数時間で動作する。

4 おわりに

本研究では、センサネットワークの自律的かつ安定的運用手法の確立を目指し、アルゴリズム設計理論の観点からは自己安定アルゴリズムの出力変更回数の削減に着目し、観測データ処理の観点からはデータの収集とストリーム処理に着目した。得られた結果はセンサネットワークのみならず、他の分散システムやデータ処理にも有用なものであり、今後はより広い分野での発展が期待される。本研究では理論的な枠組みの中での正当性、性能保証を主眼に置いたが、実際のセンサネットワークでの運用手法についてはさらなる検討や実装・実験が必要である。今後は理論的な側面、実用上の側面両方から研究をすすめる必要がある。

【参考文献】

1. N. Burri, P. von Rickenbach, and R. Wattenhofer. Dozer: Ultra-low power data gathering in sensor networks. In Proc. of IPSN 2007, pp. 250—259.
2. E.D. Demaine, A. Lopez-Ortiz, and J.I. Munro. Frequency estimation of internet packet streams with limited space. In Proc. of ESA 2002, pp.348--360.
3. J. Deng, Y.S. Han, W.B. Heinzelman, and P.K. Varshney. Balanced energy sleep scheduling scheme for high-density cluster-based sensor networks. Computer Communications, 28, pp. 1631—1642, 2005.
4. E.W. Dijkstra. Self-stabilizing systems in spite of distributed control. Communications of ACM, 17 (11), pp.643—644, 1974.
5. S. Dolev. Self-stabilization. The MIT Press, 2000.
6. S.T. Huang. A self-stabilizing algorithm for constructing breadth-first trees. Information Processing Letters, 41, pp.109-117, 1992.
7. R.M. Karp, S. Shenker, and C. Papadimitriou. A simple algorithm for finding frequent elements in streams and bags, ACM Transactions on Database Systems, 28, pp.51--55, 2003.
8. G.S. Manku, and R. Motwani. Approximate frequency counts over data streams. In Proc. of VLDB 2002, pp.346--357.

〈発表資料〉

題名	掲載誌・学会名等	発表年月
A randomized algorithm for finding frequent elements in streams using $O(\log \log N)$ space.	Proc. of 22nd International Symposium on Algorithms and Computation, pp.514—523.	2011年12月
Monotonic Stabilization.	Proc. of 14th International Conference on Principles of Distributed Systems, pp. 475—490.	2010年12月
自己安定アルゴリズムの出力変化回数削減手法の提案.	情報処理学会研究報告, Vol.2012-MPS-87, No.4, pp.1-8.	2012年3月
ストリーム中の頻出アイテム発見に対する $O(\log \log N)$ 領域乱択アルゴリズム	情報処理学会研究報告, Vol.2011-AL-137, No.1, pp.1—5.	2011年11月
無線センサネットワークでのノードの停止・追加を考慮した省電力データ収集手法.	情報処理学会研究報告, Vol.2011-MPS-82, No.7, pp.1--6	2011年3月