

ワイヤレスセンサネットワークにおける Ant Colony Optimization を用いた適用的経路探索 Adaptive Routing for Wireless Sensor Network using Ant Colony Optimization

細川 太輔† Goutam Chakraborty† 馬淵 浩司† 松原 雅文†
Daisuke Hosokawa Goutam Chakraborty Hiroshi Mabuchi Masafumi Matsuhara

1. はじめに

近年の小型センサの発達とその低価格化によって、センサネットワーク実用化にむけての研究が盛んに行われている。しかし、センサネットワークには様々な課題があり、それを解決するのは非常に困難で、今現在解決には至っていない。特に、ルーティングの問題の解決はセンサネットワーク実用化へ向けての最重要課題と言える。これを解決することでセンサネットワーク実用化が現実性を帯びてくる。

本稿ではセンサネットワークへのルーティングに ACO (Ant Colony Optimization) を用いたものを提案する[1]。ACO とは、蟻の採餌行動を基にしたメタヒューリスティクスである。メタヒューリスティクスの中でも性能がよく、また、動的な探索性能を持つという特徴がある。よって、探索状況が動的に変化するセンサネットワークの問題解決に適している。また、ルーティングの前にネットワークのクラスタリングを行うことでルーティングの効率化、およびスケーラビリティの向上を図る。さらに、クラスタリングにも ACO を用いることで、一貫した効率のよい通信アルゴリズムの作成を目指す。

2. ワイヤレスセンサネットワークに関して

2.1 ワイヤレスセンサネットワーク

センサネットワークの中でも、用いるセンサがそれぞれ無線通信機能を持つものをワイヤレスセンサネットワークと言う。ワイヤレスセンサネットワークは、センサを対象領域に自由に配置することができる半面、固定で配置するセンサよりも、通信、消費電力など考慮しなくてはならない問題が増える。また、ワイヤレスセンサネットワークは通信方式から、アドホックネットワークの側面も持つ。

2.2 ルーティングプロトコル

MANET で提案されているアドホックネットワーク向けのルーティングプロトコルの適用が考えられている。アドホックネットワークのルーティングは大きくわけて、通信の際に経路を構築する Reactive 型、常にネットワークを監視し最適な経路を保持しておく Proactive 型に2分される。しかし、どちらも単純に適用した場合、負荷が高く、ワイヤレスセンサネットワークへの適用は困難だと言われている。

2.3 ネットワークトポロジー

シンクを複数配置するマルチシンクと単一シンクのネットワークが考えられる。単一シンクはマルチシンクに比べると、シンク付近のノードへの負荷が集中し、ネットワーク寿命が短くなる問題が発生する。一方、マルチシンクは、障害に強く、ネットワークの負荷が軽減できる半面、シン

クの効率的な配置を実現することが難しい。

2.4 対象ネットワーク

提案手法を適用する対象として、広大な領域に対して、多数のセンサノードを配置し、環境モニタリングを行うような大規模ワイヤレスセンサネットワークを想定する。大規模ワイヤレスセンサネットワークにおいてはノード数が数百から数千と非常に多くのノード同士が不特定多数と通信するため、高いスケーラビリティをもった通信アルゴリズムが必要となる。また、環境モニタリングでは、長い時間人が立ち入らない、もしくは立ち入れないような環境への適用も想定されるためネットワークの長寿命化とそれに伴った軽負荷な通信も必要である。

3. 提案手法

3.1 概要

ACO を用いたマルチシンクネットワークにおける Reactive 型のルーティングを提案する。まず、マルチシンクを実現するための方法として ACO によるクラスタリングを行う。その後、ACO による経路探索、経路構築を行った後、実際に通信を行う。クラスタリングとルーティングを行うことで効率的なネットワーク運用、およびスケーラビリティの向上を目指す。また、メタヒューリスティクスを用いることで経路構築のコストを削減し、Reactive 型のルーティングを行うことで経路保持のコストの削減を目指す。

3.2 クラスタリング

クラスタリングアルゴリズムとしては Ant Clust を基にしたものを使用する[2]。Ant Clust は各エージェントごとの類似度を測定し、それをもとにして各エージェントが相互に作用し、クラスタリングを行う。クラスタリングを行う際に、クラスタ数を指定する必要がないため、Ant Clust を用いてクラスタリングを行った対象に対して、動的な再クラスタリングが可能である。またクラスタリング時に、各クラスタ内でノード同士のコストを計算し、クラスタヘッドを決定することでマルチシンクを実現する。

3.3 ルーティング

ルーティングのアルゴリズムには、Ant Colony System(ACS)を基にしたものを使用する[3]。ACS は ACO の中でも高い探索性能を持つアルゴリズムで、障害に強い動的なルーティングが可能になる。

3.4 通信手順

図 1 にフローチャートを示す。まず、ネットワークの状況を判断して、クラスタリングが必要な場合は Ant Clust を実行しクラスタを生成し、各クラスタ内においてクラスタヘッドを決定する。その後、各クラスタ内で、ACS を用いてクラスタヘッドとノード間での経路構築を行い、クラスタヘッドがクラスタ内のデータを収集する。最終的なシンクへのデータ送信は、クラスタヘッド間でデータを送信することによって行う。クラスタヘッド間での通信もクラスタ内と同様に ACS によって行う。通信が進むとクラスタ

†岩手県立大学大学院 ソフトウェア情報学研究所

ヘッドを始めとした各ノードの電力低下が起こる。その際、再クラスタリングを行うことでネットワークの状態をその都度、最適なものに保つ。

クラスタリングから経路構築までを ACO のアルゴリズムを基にして行い、クラスタ間、クラスタ内のルーティングをどちらも ACS を用いて行うことによって、ノード数の増加にも対応できる、高いスケーラビリティをもった通信が期待できる。

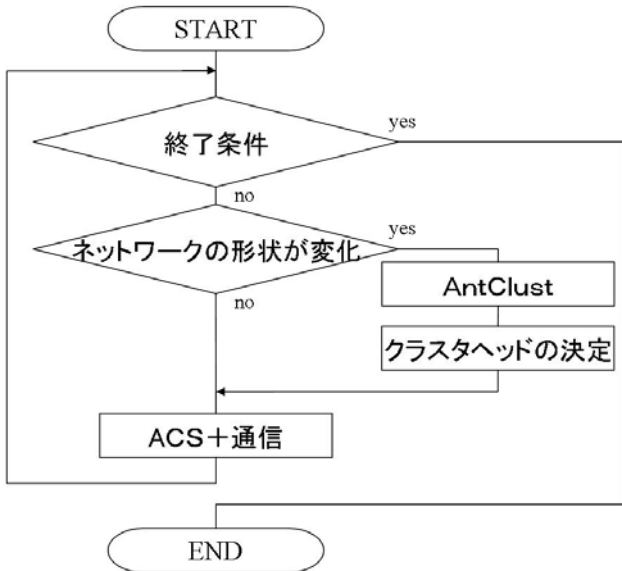


図1：通信手順

4. 評価実験

提案手法を実際のネットワーク運用に即したシミュレーションで評価する。

4.1 実験条件

600*600 のマップに 200 個のノード数を、一様分布に従って配置する。電波送信にかかる消費電力は距離の 2 乗に比例する。それぞれのノードはクラスタリング後、各クラスタのクラスタヘッドへデータを送信する。

4.2 結果と考察

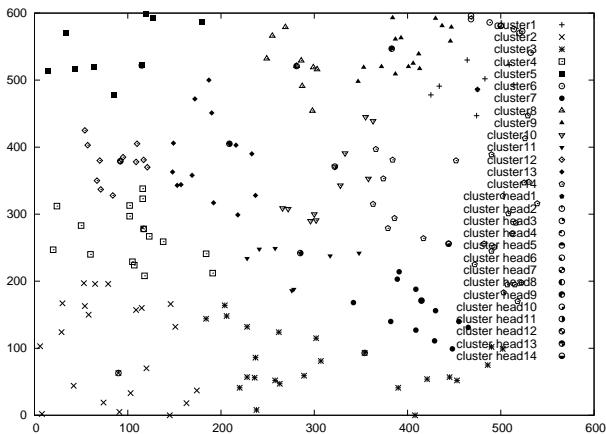


図2：クラスタリング結果

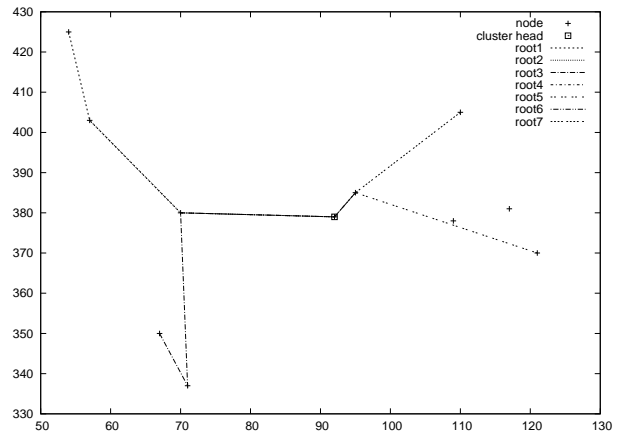


図3：クラスタ内の通信経路例

図2に Ant Clust によるクラスタリング結果を示す。クラスタ数は指定していないが、各ノード同士の類似度に基づいて、状況に応じたクラスタが動的に生成されている。ノードの配置は一様分布だが、クラスタサイズが大きくなり過ぎず、各クラスタが同程度の規模で生成されている、これは Ant Clust によるクラスタリングが有効に作用していると考えられる。さらに、各クラスタのクラスタヘッドもクラスタ内のコストに応じて自動的に決定されている。図3に ACS によるクラスタ内でのルーティングの例を示す。各クラスタ内でフェロモン値に基づいてルーティングが行われるが、それぞれのノードからクラスタヘッドに向けて比較的短い経路によって通信が行われている。各クラスタ内では同一のフェロモンを用いているため、別のノードの通信によって使用されたフェロモン値を利用して探索を行うことができる。これによって経路構築にかかる計算時間の短縮、経路構築の際の通信による無駄な電力の消費を抑えることが期待できる。

5. おわりに

通信前にクラスタリングを行うことでシンクに対して負担を軽減したルーティングを行うことができた。また、動的なクラスタリングによってネットワークの状況に応じたシンクの配置が可能となった。さらに、クラスタリングを行うことでスケーラビリティが向上し、ネットワークのサイズによらず、同等の性能が期待できる。今後は、他の手法との比較を行い、本手法の有効性を検証していく。

参考文献

- [1] 大内東, 山本雅人, 川村秀憲, 柴肇一, 高柳俊明, 當間愛晃, 遠藤聡志, "生命複雑系からの計算パラダイム—アントコロニー最適化法・DNA コンピューティング・免疫システム", 森北出版, pp. 1-76(2003).
- [2] N. Labroche, N. Monmarché, and G. Venturini, "A new clustering algorithm based on the chemical recognition system of ants," in Proceedings of ECAI 2002, pp. 345-349, July 2002.
- [3] M. Dorigo and L. M. Gambardella, "Ant colony system: Acooperative learning approach to the traveling salesman problem"; IEEE Transactions on Evolutionary Computation, 1-1, pp. 53-66(1997).