

引力・斥力モデルに基づいたセンサノードの動的再配置手法

村瀬正名¹ 西尾信彦^{1 2} 徳田英幸^{1 3}

¹ 慶應義塾大学大学院 政策・メディア研究科

² 科学技術振興事業団, さきがけ研究 21 「協調と制御」領域

³ 慶應義塾大学 環境情報学部

本稿では, モバイルセンサネットワークを対象とし, アプリケーションの期待するセンサノード配置を実現する SDSP (Self-organizing Dynamic Sensor Placement) 方式を提案する. センサネットワークではアプリケーションにとって, センサノードの位置が重要となる. アプリケーションが期待する位置にセンサノードが存在しない場合は, その目的を達成できない. しかし, センサネットワークにおける既存研究の多くは, この問題に対してほとんど考慮されていない. 本稿で提案する SDSP 方式は, 各センサノードが自律的に移動し, センサノード分布の偏りを解消する. 各センサノードは Random Waypoint モデルに基づいた移動を行い, その移動中に調査対象を探索する. 加えて, ノイズやセンサデータの不確定性, 誤検知などの問題に対しても, 複数のセンサノードが同一の対象物を監視することでセンシング精度を高める. 同一対象物に集まるセンサノードの数を制御するため, 本研究では引力・斥力モデルを導入した. 本方式をシミュレーションによって評価し, コスト(センサノード数, エネルギー消費量)とパフォーマンス(イベントロスト時間)の関係を明らかにした.

A Self-organizing Dynamic Sensor Placement Scheme Based on the Attraction/Repulsion Model

Masana Murase¹, Nobuhiko Nishio^{1 2}, and Hideyuki Tokuda^{1 3}

¹ Graduate School of Media and Governance, Keio University

² “Intelligent Cooperation and Control,” PRESTO, Japan Science and Technology Corporation(JST)

³ Faculty of Environmental Information, Keio University

This paper proposes a Self-organizing Dynamic Sensor Placement (SDSP) scheme for mobile sensor networks to realize sensor deployment as applications expect. In sensor networks, the placement of sensor directly affects the results of exploration. When sensors are not deployed to the location point as application expects, sensors may not explore the entire expected area and provide the necessary sensory data. However, conventional research in sensor networks does not cope with that problem. Our proposed scheme can solve deviation of sensor-node-distribution by sensors' proactive movement based on the Random Waypoint model. Moreover, several mobile sensors always capture data for the same target to enhance the sensing accuracy and decrease the possibility of error occurrence. To control the number of sensor nodes that observe the same target, we apply the Attraction/Repulsion model to the SDSP scheme. We had evaluated the basic performance of the SDSP scheme in terms of cost (the number of nodes and energy consumption) and performance (event lost time).

1 はじめに

近年の技術革新により, 同一のシリコン基板にセンサやアクチュエータ, 計算機能を集約できるようになった [7]. さらに, 低電力 CMOS 無線通信デバイスの登場によって, 無線通信機能と計算機能を有した小型センサの開発が行われている. U.C.Berkley をはじめとする研究機関は小型センサ, Mote[4] を開発し, 国や大学の他の研究機関ばかりでなく民間企業にも提供している.

このような小型センサ(センサノード)を数百, 数千の単位で協調動作させて情報収集を行うセンサネットワークの研究 [3], [6] が注目を集めている. 大量のセンサノードを分散配置することによって, 大規模な情報収集を可能にし, 環境モニタリング(大気, 水質, 土壌など)や軍事における索敵, 惑星調

査, 災害地調査など様々な分野に応用できる. また, ユビキタスコンピューティング環境において, ユーザの状況に適応できるコンテキストウェア・アプリケーションに対してセンサデータを提供することも可能である. 実際に大量のセンサノードを分散配置するには, 屋外の場合, 上空からセンサノードを散布する方式が効率的である. また, 屋内の場合には建材にセンサノードを埋め込むことにより大量のセンサノードを分散配置できる. 分散配置が完了すると, 各センサノードは無線通信デバイスを利用し, 取得した環境情報(例えば, 野生生物の有無や汚染物質の濃度など)のやりとりを行う.

センサネットワークにおける既存研究の多くは, センサノードが均等に分散配置されていると仮定している. しかし, 特に屋外でセンサノードを分散配

置する場合、センサノードの初期配置の分布に偏りが生じ、アプリケーションの期待する環境情報を取得できない可能性が発生する。例えば、遭難者捜索を考える。この場合、飛行機やヘリコプターを利用して、上空からセンサノードを散布することによってノードの分散配置を実現することになる。しかし、風や地形などの自然的要因により必ずしも遭難者が実際にいる場所やその付近にセンサノードが落下するとは限らない。仮にセンサノードの落下地点が遭難者の近くであっても、遭難者の位置を特定するために十分なセンシング精度が得られるとは限らない。これは、ノイズによる不確実性がデータに乗ってしまう場合や、センサ自体が高性能でないために適切なデータを取得できない場合が発生し得るためである。したがって、センサノードの配置はアプリケーションの目的達成を左右する重要な要素であると言える。

2 ノード配置問題

本節では、センサネットワークにおける問題としてノード配置問題を取り上げる。既存研究の多くはセンサノードの配置が適切に行われたことを仮定しているが、アプリケーションの要求に合わせたノード配置を実現する研究は少ない。したがって、本研究で取り組む意義は大きい。

ノード配置問題とは、アプリケーションの意図したセンサノードの配置が実現されないことであり、本研究ではノード配置問題を次の小課題の複合問題と定義する。

- 監視エリア問題
アプリケーションが監視したい空間にセンサノードが配置されない状況が発生すること。
- 精度問題
アプリケーションが期待する精度でセンサデータを取得できない状況が発生すること。

ただし、アプリケーションにとって興味ある場所をあらかじめ特定でき、人が直接全センサノードを配置するのが容易な場合はノード配置問題は発生しない。だが、大量のセンサノードを利用するセンサネットワークの応用分野においてはこうした想定は非現実的であり、ノード配置問題に対処しなければならない。

2.1 監視エリア問題

この問題は、センサノード数とイベント発生数に依存するが、イベント発生場所を特定できない場合には、調査対象空間全体にセンサノードを散布させなければならない。すなわち、 N 個のセンサノードで調査対象空間全域を監視できなければ、アプリケーションが監視したいエリアを見落としてしまう。ここで、センサノード i ($1 \leq i \leq N$) の検知可能な範囲の面積を S_{c_i} とし、調査対象空間の面積を S_{rz} とすると、

$$D = \frac{2 \sum_{i=1}^N S_{c_i}}{\pi S_{rz}} \quad (1)$$

という調査対象空間全域を N 個のセンサノードで観測可能かを判別する式が得られる。ただし、センサノードのセンシング範囲を球状と仮定する。 $D < 1$ のときには、センサノード数が不足しているため、調査対象空間全体を監視できない。また理論上は、 $D \geq 1$ の場合にセンサノードが一様に分布されて

いたら、センサノードの不足は起きないが、条件によっては調査対象空間全体を監視できない場合が発生する。特にセンサノードを上空から散布する際には、風や地形、あるいは人為的ミスによって必ずしも調査対象空間に均等にセンサノードが配置される保証がない。このため、センサノード分布に偏りが生じ、アプリケーションが監視したいエリアを見落してしまう可能性がある。

2.2 精度問題

精度問題とは、主にふたつの課題から成る。ひとつは、ノイズ (センサデータの不確実性) あるいはセンサ自体の確度が低いために生じるセンシング精度の劣化である。モバイルセンサネットワークに限らず、既存のセンサネットワークの分野においても、センサ自体のコストを下げることでより多くのセンサノードの分散散布あるいは分散配置を実現する。この経済的コスト削減による弊害としてセンサ性能の低下が挙げられ、アプリケーションによっては、センサノード単体では期待するセンサデータを取得できない場合が発生する。例えば、遭難者捜索や生態調査などで対象物の位置情報を取得したい場合、その誤差が 10m あった場合には、その対象物の居所や軌跡を正しく描けない。こうした問題に対応することの重要性は Bulusu らの指摘 [2] から窺える。

また、もうひとつの課題は誤検知である。センサノード単体で調査対象や現象を観測する場合を考える。センサノードに搭載されているセンサが高性能である (エラー発生率が低い) 場合には、誤検知による被害は少ないことが予想される。しかし、モバイルセンサネットワークや既存のセンサネットワークでは、大量のセンサノードを分散配置させることにより大規模な調査を行う。したがって、センサノード単体、特に搭載されているセンサに経済的コストをかけることは難しい。ゆえに、ノードにエラー発生率の高いセンサを搭載することになり、誤検知の発生は免れないと予測できる。

3 モバイルセンサネットワーク

第 2 節で示したノード配置問題を解決するため、センサノードが自律的に移動できるモバイルセンサネットワークを提案する。センサノードの自律的な移動は、監視エリア問題を解決するだけでなく、精度問題にも有効な対応策となり得る。監視エリア問題に対しては判別式 1 の値が 1 より小さい場合でも各センサノードの移動によって監視エリア問題の発生回数を軽減できる。また、精度問題に対しても複数のセンサノードが同じ対象物あるいは現象を監視し、センサフュージョンの技術 [9], [8] を利用することでセンシング精度を向上できる。

モバイルセンサネットワークは、複数の移動可能なセンサノード (モバイルセンサノード) によって構成され、既存のセンサネットワーク同様マルチホップ無線網を構築する (図 1)。また、有線基盤網との接続性を確保するためにシンクノードも存在する。本研究ではシンクノードへのセンサデータへの転送は既存のルーティング技術 [1], [11] を用いる。

モバイルセンサノードには、自走型、飛行型、航行型が存在し、環境や用途に応じたノードを選択できる。モバイルセンサノードの通信デバイスおよびセンサデバイスには有効範囲 (R_c : 通信可能範囲, R_s : センシング範囲) が存在し、図 1-(b) ($R_c > R_s$)

に示す関係にある。例えば、IEEE802.11b 準拠の無線デバイスでは見通し 100m の範囲で通信可能なに対して、センサ (例えば人感センサ) は 10m から 50m が検知可能範囲である。この関係より、判別式 $1: D \geq 1$ を満足させるためには、大量のモバイルセンサノードを必要になり、十分な数のセンサノードが存在する仮定は現実的ではない。したがって、判別式 $1: D < 1$ の状況に対応したモバイルセンサノードの再配置が重要となる。

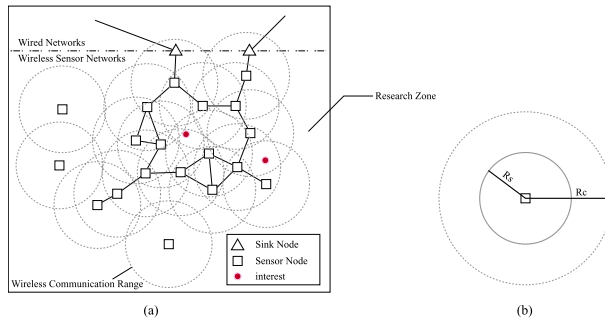


図 1: モバイルセンサネットワーク環境

本稿では、モバイルセンサノードの通信デバイスやセンサデバイスの性能および移動能力は同質であると仮定する。また、全モバイルセンサノードは GPS などの位置情報特定デバイスが搭載されているか、Localization 技術によって常に自身の位置情報を取得できると仮定する。この仮定は、携帯電話に GPS が低コストで搭載されるようになった社会背景から鑑みても現実的と言える。

モバイルセンサネットワークにおけるアプリケーションは、調査対象空間 (Research Zone) の境界情報を保持する。各モバイルセンサノードはこの境界情報と自身が取得する位置情報によって、調査対象空間外に移動することを抑制する。また、アプリケーションは興味ある環境情報を interest として保持し、この interest にしたがって各モバイルセンサノードの配置を動的に変化させる。本稿では interest を下記のように定義する。

「センサによって取得された数値情報がアプリケーションの指定する値より大きいか、小さいあるいは同じイベント」

interest 情報の具体例として、「人や生物の有無」や「 $10pg - TEQ/m^3$ を超えるダイオキシン濃度の場所および状況」などが挙げられる。

4 関連研究

本研究の関連研究として、自走型センサノード (Robotics) を応用し、センサの動的分散配置を可能にする [12], [5], [10] を参照する。

4.1 Distributed Sensing and Data Collection Via Broken Ad hoc Wireless Sensor Networks of Mobile Robots

Winfield は、ランダム歩行アルゴリズムによるモバイルセンサノードの拡散およびランダム歩行時の適切なルーティングアルゴリズムを提案している [12]。各モバイルセンサノードは無線通信デバイスを搭載し、取得したセンサデータをやりとりする。彼は、各モバイルセンサノードをランダム歩行させ

ることで実現される分散データ収集に着目し、さらにその際に生じるネットワークの分断問題に対処している。モバイルセンサノードの移動によりネットワークポロジが動的に変化するためシンクノードまでセンサデータをルーティングできない可能性が生じる。そこで、近隣 (無線通信範囲で 1 ホップ以内) に存在する他のモバイルセンサノード情報をビーコンあるいは制御パケットを利用して取得する。この情報をもとに、新たな近傍ノードを発見すると、センサデータをそのノードに宣伝する。この操作を繰り返すことでネットワーク全体にセンサデータを行き渡らせる。[12] ではランダム歩行アルゴリズムによるモバイルセンサノードの拡散および拡散中の適切なルーティングを行え、これによって監視エリア問題に対応できる。しかし、各モバイルセンサノードは常に移動し続けるためノードのエネルギー損失が大きい。また、本研究が対象とする精度問題に対しては考慮されていない。

4.2 An Incremental Self-Deployment Algorithm for Mobile Sensor Networks

Howard らは、モバイルセンサノードの拡散手法として Incremental Deployment アルゴリズムを提案している [5]。彼らは、モバイルセンサノードのセンシング範囲の合計値を最大化することを目指している。Incremental Deployment アルゴリズムでは、先に移動にしたセンサノードから移動場所決定に必要な情報 (未監視領域など) を無線通信により取得し、各モバイルセンサノードのセンシング範囲が重複しないように移動する。ここで注視すべき点は、障害物を考慮していることと調査対象空間が未知でも動作することである。各モバイルセンサノードは移動中に障害物情報やマップ情報を生成し、次に移動するモバイルセンサノードに通知する。このアルゴリズムによって監視エリア問題に対処できる。しかし、この方式を利用する場合、モバイルセンサノードの拡散は逐次的にしか行われぬ。モバイルセンサノードの数が増加すると拡散完了までにかかる時間は理論上線形に増加することになる。したがって、アプリケーションによってはこの時間的損失は致命的な結果をもたらす可能性がある。また、本研究が対象とする精度問題へは未対応である。

4.3 Cooperative Robotics for Multi-Target Observation

Parker は、他のモバイルセンサノードとの位置関係を利用し、各モバイルセンサノードが同じ対象物を同時に監視しないように移動する方式 ALLIANCE を提案している [10]。この方式では、センシング範囲を考慮し、各モバイルセンサノードは他のモバイルセンサノードのセンシング範囲内に移動できないアルゴリズムを考案している。各モバイルセンサノードは互いの位置情報を交換し、その情報をもとに移動位置を決定する。これによって、センシング範囲の重なりを減らし、センシング範囲の合計値を最大にでき、対象物の探索領域を広げることが可能である。また、ALLIANCE 方式では対象物 o がモバイルセンサノード i のセンシング範囲内に存在する確率および、 i 以外の他のモバイルセンサノードが対象物 o を既に監視している確率を組み合わせ、さらにセンシング範囲の重なりを軽減している。以上により、監視エリア問題に対処することは可能である。しかし、本研究が対象とするセンシング精度

問題には、未対応である。特に、ALLIANCEは単一のモバイルセンサノードによって対象物を観測しようとしているが、この場合センサの誤検知に対応できない。したがって、複数のモバイルセンサノードが協調動作し、誤検知への対応や不確定性の軽減などは重要な課題と言える。

5 引力・斥力モデル

本稿において、モバイルセンサノードの動的再配置は「引力・斥力モデル」に基づいて処理される。引力・斥力モデルとは図2に示した力場 (Force Field) を構成し、他のモバイルセンサノードを引き寄せたり、逆に引き離したりする。引力・斥力の大きさを調整することで、集まるモバイルセンサノード数を調整できる。ここで、センサノードによって引き寄せられたノードを「最近傍ノード (Nearest Neighbor Node)」と呼ぶ。

図2において力場の形状は円形で表現されているが、その形状に制限はない。したがって、楕円状の力場や矩形の力場も存在し得る。また、力場は複数存在しても構わず、interestを発見したモバイルセンサノードの数だけ力場は存在する。加えて、複数の interest が近い位置に存在する場合には、力場が重なる可能性がある。

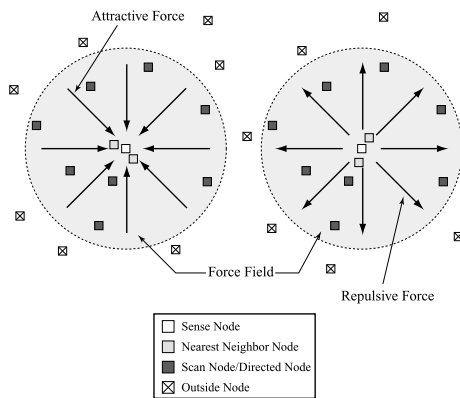


図 2: 引力・斥力モデルの概念図

本モデルでは、力場の有効範囲を力の大きさによって表現する。したがって、引力や斥力の値 (絶対値) の増加に伴い、力場の有効範囲も広がる。逆に、引力や斥力の値 (絶対値) が小さくなれば、力場有効範囲は狭くなる。力場の有効範囲は時間によって変化し、アプリケーションの期待するセンシング精度 (あるいはセンシング精度を向上するために十分な数のセンサノード数) に達成すると消滅する¹。次節において、引力・斥力モデルを利用した引力・斥力制御方式について述べる。

6 モバイルセンサノードの動的再配置

本節では、モバイルセンサノードの動的再配置手法である SDSP (Self-organizing Dynamic Sensor Placement) 方式について詳説する。

6.1 SDSP 基本動作

図3はSDSP方式の状態遷移図を表している。各モバイルセンサノードは図のように *SCAN_MODE*, *SENSE_MODE*,

¹ 引力・斥力がゼロであることを意味する。

DIRECTED_MODE の三つのモードを遷移する。アプリケーションが起動すると、各モバイルセンサノードは *SCAN_MODE* として動作する。*SCAN_MODE* は、調査対象空間内にて interest 発見を行うモードである。interestを発見すると、*SENSE_MODE* に移行し、その地点に留まって観測を続ける。ここで、interestを見失うと *SCAN_MODE* から *SCAN_MODE* へと遷移し、再び新たな interest を探索する。

また、*SENSE_MODE* のモバイルセンサノードは、引力メッセージ (本稿では、モバイルセンサノードが送受信するビーコンあるいはパケットのことをメッセージと呼ぶ) を受信することで、*DIRECTED_MODE* に移行する。*DIRECTED_MODE* では、メッセージ内の移動位置情報を利用して定められた地点に移動する。*DIRECTED_MODE* のモバイルセンサノードが斥力メッセージを受信すると、再び *SCAN_MODE* に遷移し、新たな interest を探索する。

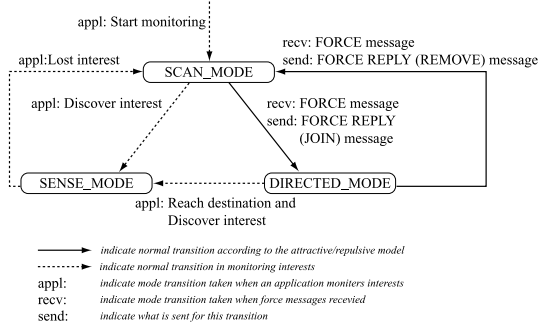


図 3: 状態遷移図

6.1.1 Scan Mode

SCAN_MODE のモバイルセンサノード (スキャンノード) は動的ランダム拡散方式に基づいた移動を行う。動的ランダム拡散方式として変則 Random Waypoint モデルを採用する。変則 Random Waypoint モデルとは、任意の移動先を決定し、Random Waypoint モデルと異なり速度はランダムに決定せず、一定で等速直線運動するモデルである。通常の Random Waypoint モデルのように目的地に到達すると一定時間停止し、この時間経過後に再び移動地点を一様分布に基づいて決定する。モバイルセンサノードは、この移動中に搭載されているセンサを利用して、interest を探索する。

6.1.2 Sense Mode

SENSE_MODE に移行したモバイルセンサノード (センスノード) は主にふたつの処理を行う。ひとつは現在地点での interest 観測であり、もうひとつはセンシング精度を向上のための処理である。センシング精度向上のためにモバイルセンサノードの動的再配置を行う。動的再配置を行うために本研究では第5節に示した引力・斥力モデルを導入する。

6.1.3 Directed Mode

DIRECTED_MODE のモバイルセンサノード (ディレクティッドノード) は、引力メッセージ内の移動位置情報を抽出し、その位置に移動するモードである。このモードで動作しているモバイルセンサノード

は, interest 探索は行わず, 指定された位置に移動した後に interest を観測する. *DIRECTED_MODE* から *SENSE_MODE* に移行したモバイルセンサノードは, その状態遷移情報も保持する.

6.2 引力・斥力モデルによる動的再配置

引力・斥力制御方式の基本動作を図4に示す. センスノードは, まず力場の適用範囲を算出し, 引力メッセージか斥力メッセージを作成する. 本稿では, 引力メッセージ, 斥力メッセージを総称して *FORCE* メッセージと呼ぶ. 次に, そのメッセージを無線通信デバイスを用いて送信する. メッセージが引力の場合, そのメッセージを受信したスキャンノードはランダム時間待ち(ただし, この間ノードは移動している), 再び同一センスノードからメッセージの届く範囲にいるか判断する. この判定が真の場合には, 引力メッセージを処理し, そのメッセージによって指定された位置へ移動を開始する.

一方, メッセージが斥力の場合, そのメッセージを受信したディレクティッドノードはランダム時間待ち後に *FORCE REPLY* メッセージを送信し, *SCAN_MODE* に遷移する. また, 斥力メッセージを受信したノードがスキャンノードの場合には, そのメッセージを發したセンスノードが観測している interest を捕捉しないように移動する. 斥力メッセージは, それを發したセンスノードにとっての最近傍ノード数が過多であることを示唆している. したがって, これ以上同じ interest に他のモバイルセンサノードを集結させる必要はない.

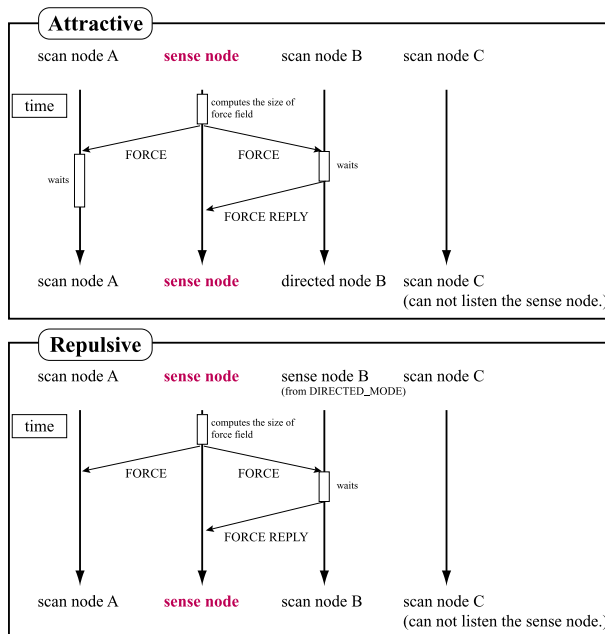


図 4: 引力・斥力制御方式の基本動作

6.2.1 ユークリッド距離を利用した力場

距離による算出方法では, センスノードからの距離情報を利用して力場の有効範囲を決定する. 図5-(a)に示すように, 引力あるいは斥力が小さい場合にはセンスノードから距離が近いノードのみ力場の影響を受ける. 一方, 引力あるいは斥力が大きく

なると, センスノードから離れたノードに対しても力場による影響を受けるようになる(図5-(b)).

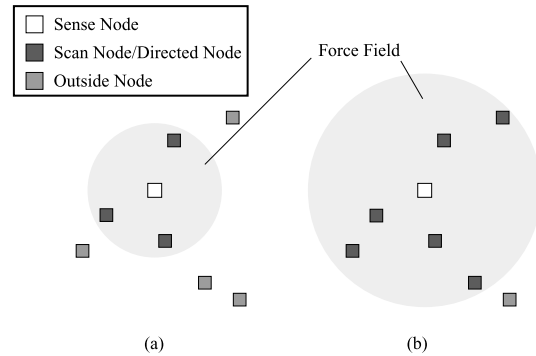


図 5: ユークリッド距離による力場算出方法

本算出方法では, 力場の有効範囲(有効距離)を ∞ まで定義できるが, 計算量を減らすため, 最大有効範囲(最大有効距離)を無線通信範囲 R_c と定める. この距離による算出方法 *COMPUTEDISTANCEBASEDFORCELEVEL* は図6のようになる.

```

COMPUTEDISTANCEBASEDFORCELEVEL()
    level ← (Rc/required_node_num) ×
    (required_node_num - countNearestNeighborNodes())
    if level < -1 × required_node_num then
        level ← -1 × required_node_num
    return level

```

図 6: *ComputeDistanceBasedForceLevel()* の動作

手続き *COUNTNEARESTNEIGHBORNODES()* によって最近傍ノード数を調べ, その情報をもとに力場の適用範囲を決定する. *if* 文による再計算は, 力場有効範囲が無線通信可能範囲 R_c を超えないようにするための処理である.

受信時の処理は, *FORCE* メッセージ内に含まれる *LEVEL* フィールドの値を利用し, 下記の条件によって力場有効範囲内に自身が存在するか調べばよい.

$$L_{own} - L_s \leq LEVEL \quad (2)$$

ただし, L_{own} は *FORCE* メッセージを受信したモバイルセンサノードの位置情報を表し, L_s は *FORCE* メッセージを發したセンスノードの位置情報を意味する. この位置情報同士の減算結果は, 2点間のユークリッド距離と定義する.

6.2.2 移動位置決定方法

移動位置決定方法は, 最初の移動位置をセンスノードが決定するか移動すべきディレクティッドノードが決定する方法の二種類が考えられる. 前者の方法では, まずセンスノードが自身の位置から R_s の範囲内においてランダムに移動位置を決定し, この情報をディレクティッドノードに通知する. 次に, ディレクティッドノードがその位置に移動し, かつ interest を補足できた場合にそのノードはセンスノ

ドの最近傍として登録される(ディレクティブドノードがセンスノードに移動完了を通知する)。移動完了後に interest を発見できなかった場合には、ディレクティブドノードがセンスノードから R_s 範囲内で移動位置をランダムに定め、移動を繰り返すことで最終的に interest を発見できる。

一方、ディレクティブドノードが最初の移動位置を決定する方法も基本的な動作は前者の方法と違はない。最初の移動位置を受信した FORCE メッセージに含まれるセンスノード位置情報を利用し、その位置から R_s の範囲内でランダムに移動位置を決定する。

6.2.3 力場選択方法

複数のセンスノードから FORCE メッセージを受信した場合は、力場選択処理を行わなければならない。通常、通信に無線媒体を利用した場合には、媒体アクセス制御技術によって、あるひとつのノードからしか電波は届かない。IEEE802.11b の場合には、CSMA 方式による媒体アクセス制御を行っている。したがって、スキャンノードは、完全に同時には複数の FORCE メッセージを受信しない。本研究では、この時間差を利用することで力場をひとつに選択する。

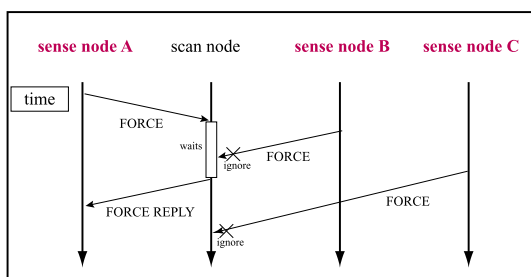


図 7: 力場選択シーケンス

図 7 に力場選択のシーケンス図を示す。スキャンノードは、センスノード A の FORCE メッセージを最初に受信した場合に、そのメッセージ処理を優先的に行う。本節のはじめに解説したように、スキャンノードはランダム時間待ち、その後 FORCE REPLY メッセージをセンスノード A に送信する。このランダム待ち時間中にセンスノード B から新たな FORCE メッセージが届いても、スキャンノードはこのメッセージを無視する。ランダム時間後にセンスノード A の通信可能範囲を超えてしまった場合、FORCE REPLY メッセージをセンスノード A に送信しない。この状況が発生すると、スキャンノードはセンスノード C から送信された FORCE メッセージを処理し、ランダム時間待ち FORCE REPLY メッセージを送信する。以上のようにランダム待ち時間内に他のセンスノードからメッセージが届いてもそのメッセージを破棄することで、力場をひとつに選択できる。

7 シミュレーション評価

本シミュレーション評価では、ns-2(Network Simulator Version 2) の拡張モジュールとして SDSP 方式を実装して以下の 2 項目の評価を行う。

評価 1 モバイルセンサネットワークにおけるコストパフォーマンス評価

評価 2 センサネットワークとモバイルセンサネットワークとの比較

本シミュレーションでは、イベント発生モデルとして一般的なポアソン分布に基づくイベント発生モデルを利用する。イベント発生確率は $5.0 \cdot 10^{-6}$ 個 / msec とし、そのイベントの持続時間を 1000sec に設定して評価を行った。また、シミュレーション環境を表 1 のように設定した。

7.1 評価 1: コストパフォーマンス

シミュレーション評価 1 では、コスト(ノード数、移動に伴うエネルギー消費)とパフォーマンス(平均イベントロスト時間)との関係を調べることを目的とする。これによって、ユーザにモバイルセンサネットワーク利用時におけるコストパフォーマンスの指標を提示する。

通常、モバイルセンサノード i が移動に伴うエネルギー消費量 E_i とすると、 $E_i = \alpha m_i v_i t_i$ を利用することで計算可能である。 α は動摩擦係数を表し、 m_i はモバイルセンサノードの質量であり、 v_i はその移動速度である。また t_i は移動に要した時間を表す。ここで αm_i は、摩擦面が同じ素材であり各モバイルセンサノードは同種のノードであることから定数として扱える。また、この値は移動速度や移動時間に依存しない値であることから無視することが可能である。したがって、モバイルセンサノード i のエネルギー消費は E_i を新たに式 3 に定めて計算できる。

$$E_i = v_i t_i \quad (3)$$

ただし、 $v_i = v_j (\forall i, \forall j)$ である。本シミュレーションでは、累積エネルギー消費量 $\sum_{i=1}^n E_i$ を利用する。

また、イベントロスト時間とは図 8 に示す時間を指す。すなわち、イベントを interest として認識したモバイルセンサノード数が十分な数に達した時点でイベントロスト時間が終了する。イベントロスト時間の最大値はイベント TTL であり、この場合イベントは発見できなかったことを意味する。本シミュレーションでは、各イベント毎にイベントロスト時間を計測し、その平均イベントロスト時間を評価軸に定める。

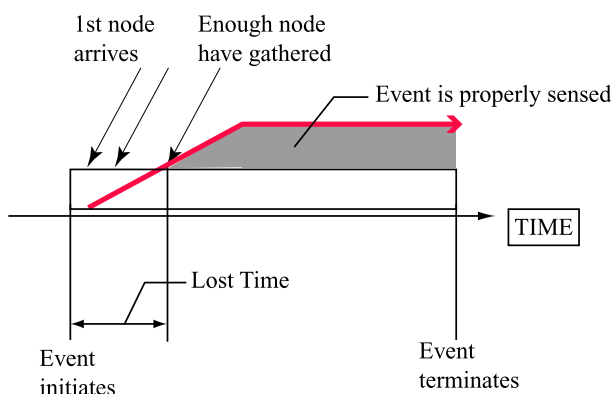


図 8: イベントロスト時間

本シミュレーションではモバイルセンサノードの移動パターンとイベント発生パターンを変えて評価

表 1: パラメータ設定 (1)

パラメータ	設定値
フィールドサイズ	1000m × 1000m
シミュレーション時間	7200sec
測定回数	各条件において 150 回
モバイルセンサネットワーク	
ノード数	100 台, 150 台, 200 台
移動速度	0.5m/sec, 1m/sec, 2.5m/sec, 5m/sec, 10m/sec, 15m/sec ^{1 2} , 20m/sec ¹ , 30m/sec ¹
センサネットワーク	
ノード数	600 台

¹ モバイルセンサノード数 100 台の場合にはこの速度でも測定

² モバイルセンサノード数 150 台の場合にはこの速度でも測定

を行った。本シミュレーション評価では 150 通りのパターンを試行した。

シミュレーション結果とその考察

図 9 にコストパフォーマンス評価の結果を示す。x 軸はモバイルセンサネットワーク全体が消費するエネルギー量を表し、y 軸は平均イベントロスト時間である。本グラフによって、モバイルセンサノード数を 100 台、150 台、200 台と変動させたときのエネルギー消費量と平均イベントロスト時間の関係を示せた。移動を行うことで平均イベントロスト時間を 1000sec 未満に抑えられた。これは、変則 Random Waypoint 移動および引力・斥力制御方式によって interest 観測に十分な数のセンサノードを集められたことを意味する。本シミュレーションでは引力・斥力決定方式には距離による算出方法を選択した。

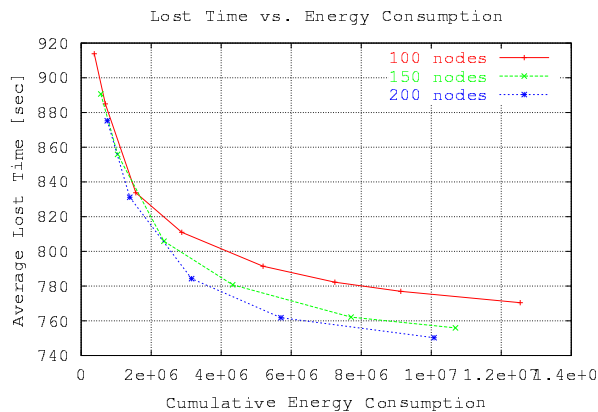


図 9: コストパフォーマンス評価結果

本シミュレーション結果より、モバイルセンサネットワーク全体が消費するエネルギー量が同じとき、必ずしもモバイルセンサノード数(ノード密度)を増加させるだけではパフォーマンスを向上できないことがわかる。これは、各モバイルセンサノードの移動速度が低速であるため、センサノードが interest を発見してから十分な数のセンサノード数に達するまでにかかる時間はモバイルセンサノード数に係わらずほとんど変わらないためである。したがって、モバイルセンサネットワークに与えられるエネルギー

量 E が $2.0 \cdot 10^6$ 以下の場合には、モバイルセンサノード数は少なくてもよい(100 台)。一方、モバイルセンサノードに与えられるエネルギー量がそれより大きい場合には、モバイルセンサノード数を増やすことで、さらにイベントロスト時間を軽減できる。

7.2 評価 2: 既存のセンサネットワークとの比較

既存のセンサネットワークとの比較を行うため、移動不可能なセンサノードを調査対象空間にランダム散布した場合のコストパフォーマンス評価を行った。既存のセンサネットワークにおいてイベントロスト時間を減らすには、ノード数を増やし、ノード密度を増加させる必要がある。本シミュレーション評価では、両センサネットワーク構築に要する経済的コストを同規模にするため、モバイルセンサネットワークのノード数を 100 台、既存センサネットワークにおけるノード数を 600 台に設定した²。ノードの初期配置は一様分布に基づいて決定した。

シミュレーション結果とその考察

表 2 にそれぞれの結果を示す。センサネットワークでは、各センサノードは移動機能を有していないため、interest 探索など移動に伴うエネルギー消費は発生しない。しかし、シミュレーション結果より平均イベントロスト時間は 1000sec とその時間を減らすことはできなかった。これは、モバイルセンサネットワークに対して 6 倍のノード密度を実現し分散配置しても、interest 発見に十分な数(本シミュレーションでは 5 台)のセンサノードが集まらないためである。すなわち、さらにノード密度を高めなければ、モバイルセンサノード 100 台と同等のパフォーマンスは得られない。

一方、モバイルセンサネットワークでは、移動速度が増加するにつれてエネルギー消費量は増加してしまうが、平均イベントロスト時間をセンサネットワークと比べて最低約 108%改善できた。このことから、本研究が提案する引力・斥力制御方式およびモバイルセンサネットワークの優位性を示せた。

しかし、モバイルセンサネットワークでは interest 探索やセンシング精度を向上させるために各ノードが移動し、バッテリーを消費する。したがって、ユーザはこのトレードオフを考慮して既存のセンサネッ

² 現在、センサノードとモバイルセンサノードの価格比は 1:6 である。

表 2: センサネットワーク vs. モバイルセンサネットワーク

ノード数	平均イベントロスト時間	平均エネルギー消費量
センサネットワーク		
600 台	1000 sec	0
モバイルセンサネットワーク		
100 台	913.8326 sec	374456.5 (0.5m/sec の場合)
100 台	791.4482 sec	5197180 (10m/sec の場合)

トワークかモバイルセンサネットワークを選択すべきである。

8 まとめと今後の課題

本稿は、アプリケーションにとって興味ある環境情報を取得し、かつアプリケーションの要求する精度でデータを取得することを目的とし、その目的達成のためモバイルセンサノードの再配置方式 SDSP 方式を提案した。本方式によって従来のセンサネットワークでは解決が困難であったノード配置問題(監視エリア問題と精度問題の複合問題)を解決できた。

監視エリア問題は、センサノードの初期配置あるいは数およびセンシング範囲によっては interest を発見・観測できない問題である。この問題に対し、SDSP 方式ではモバイルセンサノードが自律的に interest 探索を行い、センサノードの初期配置分布に係わらず、監視エリアを広げられる手法を提案した。

一方、精度問題はノイズによるセンサデータの不確定性の発生と誤検知が生じる問題である。この精度問題に対しては引力・斥力モデルを導入し、interest に集中するモバイルセンサノードの数を制御し、冗長性を高めることで対応した。また、冗長度を高め、センサフュージョンに見られる手法を組み合わせることでセンシング精度を向上できる。

加えて、本稿ではシミュレーションによって SDSP 方式の性能特性を調べた。性能特性評価では、SDSP 方式のコスト(リソースとエネルギー)と性能との比較を行い、ユーザにアプリケーション実行時の一指標を提示した。また、既存のセンサネットワークとの比較も行い、エネルギーコスト比較においては本研究は既存センサネットワークより劣るものの、経済的コストの比較およびノード配置問題、特に精度問題への対応においては本研究の方が優位であった。平均イベントロスト時間に関しては、本研究によって 108%以上の改善が見られた。

今後の課題として、実機(例えば、AmigoBot)への実装による検証を行い、SDSP 方式の実用性を評価する。また SDSP 方式のパラメータの最適値をシミュレーション評価(コストパフォーマンス評価)、実機での検証により求める。

参考文献

[1] BROCH, J., JOHNSON, D., AND MALTZ, D. The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks, 2002.
 [2] BULUSU, N., HEIDEMANN, J., AND ESTRIN, D. Adaptive Beacon Placement. In *Proceedings of the 21st International Conference on Distributed Computing Systems* (2001), pp. 489–498.

[3] ESTRIN, D., GOVINDAN, R., HEIDEMANN, J., AND KUMAR, S. Next Century Challenges: Scalable Coordination in Sensor Networks. In *ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom)* (1999), pp. 263–270.
 [4] HILL, J., AND CULLER, D. A wireless embedded sensor architecture for system-level optimization. In *UC Berkeley Technical Report* (2002).
 [5] HOWARD, A., MATARIĆ, M. J., AND SUKHATME, G. S. An Incremental Self-Deployment Algorithm for Mobile Sensor Networks. In *Autonomous Robots Special Issue on Intelligent Embedded Systems (To Appear)* (2002).
 [6] KAHN, J. M., KATZ, R. H., AND PISTER, K. S. J. Next Century Challenges: Mobile Networking for "Smart Dust". In *ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom)* (1999), pp. 271–278.
 [7] MEMS EXCHANGE. MEMS Clearinghouse. <http://www.memsnet.org>.
 [8] NAKAMURA, Y. Geometrical Fusion: Minimizing Uncertainty Ellipsoid Volumes. In *Data Fusion in Robotics and Machine Intelligence* (1992), M. A. Abidi and R. C. Gonzalez, Eds., pp. 457–479.
 [9] NAKAMURA, Y., AND XU, Y. Geometrical Fusion Method for Multi-Sensor Robotic Systems. In *IEEE International Conference on Robotics and Automation* (1989), pp. 668–673.
 [10] PARKER, L. E. Cooperative Robotics for Multi-Target Observation. *Intelligent Automation and Soft Computing* 5, 1 (1999), 5–19.
 [11] PERKINS, C., ROYER, E., AND DAS, S. Ad Hoc On Demand Distance Vector (AODV) Routing, 2002.
 [12] WINFIELD, A. F. Distributed Sensing and Data Collection Via Broken Ad Hoc Wireless Connected Networks of Mobile Robots. In *Distributed Autonomous Robotics Systems* (2000), L. E. Parker, G. W. Berkey, and J. Barhen, Eds., vol. 4, pp. 273–282.