

修士論文 2002 年度 (平成 14 年度)

モバイルセンサネットワークにおける  
ノード配置に関する研究

慶應義塾大学大学院 政策・メディア研究科

村瀬 正名

# 修士論文要旨 2002 年度 (平成 14 年度)

## モバイルセンサネットワークにおける ノード配置に関する研究

### 論文要旨

近年、プロセッサやメモリ、無線技術の革新により無線通信機能を有した小型センサノードが実現しつつある。これらのセンサノードを大量に散布・分散配置することで、生態調査や汚染物質調査など大規模な情報収集が行える。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。こうした状況は風、地形、散布方法によってセンサノードの分布に偏りが生じることで起こり得る。本論文ではこの問題をノード配置問題と呼ぶ。

本研究は、この課題を解決することを目的とし、センサノードを動的に再配置させるモバイルセンサネットワークを提案する。本論文において、モバイルセンサネットワークにおけるノードの再配置手法 SDSP (Self-organizing Dynamic Sensor Placement) 方式を設計・実装した。本方式では、センサノードの分布の偏りを解消し、調査対象空間全体を監視可能にするために、各モバイルセンサノードは変則 Random Waypoint 移動 (移動速度は常に一定) に基づいて移動する。この移動中にアプリケーションにとって興味あるデータを探索する。また、センシング精度を向上させるため、SDSP 方式では単一のモバイルセンサノードによってある対象物を観測しない。複数のモバイルセンサノードが同一の対象物を監視し、冗長性を高めることでノイズやセンサデータの不確定性さらに誤検知に対応する。

本方式を ns-2 (Network Simulator version 2) 用のシミュレーションモジュールとして実装し、シミュレーションによって本方式を評価した。本論文では、まず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測定を行った。本評価ではモバイルセンサノード数、モバイルセンサネットワーク全体で消費するエネルギー量、イベントロスト時間との関係を明らかにした。これによって、モバイルセンサノード数を単純に増加させても、イベントロスト時間を軽減できない場合が生じることがわかった。また、既存のセンサネットワークとのコストパフォーマンス比較を行った。シミュレーション結果より既存のセンサネットワークと比較してモバイルセンサネットワークは移動によるエネルギー消費を要するものの、イベントロスト時間を最低約 108% 改善できた。

キーワード:

1 センサネットワーク, 2 自律システム, 3 移動性, 4 無線ネットワーク

慶應義塾大学大学院 政策・メディア研究科

村瀬 正名

# **Abstract of Master's Thesis**

## **Academic Year 2002**

### **Self-organizing Dynamic Sensor Placement Scheme for Mobile Sensor Networks**

#### **Summary**

In recent years, advances in processor, memory and wireless technology enable small sensor nodes with capability of wireless communication. Mass deployment of these sensor nodes allows us to monitor habitat of wild life and environmental pollution level in wide range of area, which is called sensor networks. 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. This may be caused by wind, terrain and deployment methods. We call this problem as the "sensor placement problem".

This research proposes mobile sensor networks to cope with the sensor placement problem. Unlike conventional sensor networks, each sensor node dynamically changes its location during exploration in mobile sensor networks. In this thesis, we have designed and implemented the Self-organizing Dynamic Sensor Placement (SDSP) scheme to realize the mobile sensor networks.

This scheme solves deviation of sensor-node-distribution by mobile sensors' movement. In our scheme, mobile sensor nodes move according to the anomalous Random Waypoint model: mobile sensor nodes travel at a constant velocity. Moreover, several mobile sensors always capture data for the same target to enhance sensing accuracy and decrease the possibility of error occurrence.

The SDSP scheme has been implemented as a module of Network Simulator version 2 (ns-2), and evaluated by simulation. We first analyze the relationship of cost (numbers of nodes and energy consumption) vs. performance (event lost time) in mobile sensor networks. This evaluation revealed us that increasing numbers of nodes does not always enhance the performance of mobile sensor networks. In addition, we compare mobile sensor networks with conventional sensor networks in terms of cost and its performance. As a result, mobile sensor networks may consume more energy from mobility cost, its performance improved at least 108% compared with conventional sensor networks.

Keywords :

1 Sensor Networks, 2 Proactive System, 3 Mobility, 4 Wireless Networks

**Keio University Graduate School of Media and Governance**

**Masana Murase**

# 目次

第1章	序論	1
1.1	研究背景	2
1.2	本研究の目的	3
1.3	本論文の構成	3
第2章	モバイルセンサネットワーク	5
2.1	センサネットワークの概要	6
2.2	ノード配置問題	8
2.2.1	監視エリア問題	9
2.2.2	精度問題	9
2.3	研究アプローチ	10
2.3.1	モバイルセンサネットワーク	10
2.3.2	センサネットワーク vs. モバイルセンサネットワーク	11
第3章	モバイルセンサノードの配置	13
3.1	動的拡散手法	14
3.1.1	動的幾何拡散方式	14
3.1.2	動的ニューラル拡散方式	15
3.1.3	動的ランダム拡散方式	15
3.1.4	各動的拡散手法の比較	16
3.2	動的再配置手法	17
3.3	関連研究	18
3.3.1	Distributed Sensing and Data Collection Via Broken Ad hoc Wireless Sensor Networks of Mobile Robots	18
3.3.2	An Incremental Self-Deployment Algorithm for Mobile Sensor Networks	19
3.3.3	Cooperative Robotics for Multi-Target Observation	19
第4章	SDSP の設計	20
4.1	SDSP の基本動作	21
4.1.1	Scan Mode	21
4.1.2	Sense Mode	22
4.1.3	Directed Mode	22

4.1.4	引力・斥力モデルの導入 . . . . .	22
4.2	引力・斥力制御方式の設計 . . . . .	23
4.2.1	SDSP メッセージ . . . . .	24
4.2.2	引力・斥力決定方法 . . . . .	25
4.2.3	移動位置決定方法 . . . . .	30
4.2.4	力場選択方法 . . . . .	31
4.2.5	冗長問題 . . . . .	32
4.3	イベント発生モデル . . . . .	33
<b>第 5 章</b>	<b>シミュレーション</b>	<b>35</b>
5.1	シミュレーションモジュールの実装 . . . . .	36
5.2	シミュレーションモデル . . . . .	37
5.2.1	無線モデル . . . . .	38
5.2.2	センサモデル . . . . .	39
5.2.3	イベント発生モデル . . . . .	40
5.3	シミュレーション評価 . . . . .	40
5.3.1	評価 1: コストパフォーマンス . . . . .	41
5.3.2	評価 2: 既存のセンサネットワークとの比較 . . . . .	43
5.3.3	評価 3: 引力・斥力決定方式の比較 . . . . .	44
<b>第 6 章</b>	<b>結論</b>	<b>46</b>
6.1	今後の課題 . . . . .	47
6.2	まとめ . . . . .	47

# 目次

1.1	無線アドホック・センサプラットフォーム：Mote	2
2.1	センサネットワーク環境	7
2.2	モバイルセンサネットワーク環境	11
3.1	動的幾何拡散方式	14
3.2	動的ニューラル拡散方式	15
3.3	動的ランダム拡散方式	16
4.1	状態遷移図	21
4.2	引力・斥力モデルの概念図	23
4.3	引力・斥力制御方式の基本動作	24
4.4	FORCE メッセージ / FORCE REPLY メッセージ	25
4.5	HELLO メッセージ / HELLO REPLY メッセージ	26
4.6	ホップ数による力場の有効範囲設定	26
4.7	ComputeHopBasedForceLevel() の動作	27
4.8	HopRepulsion() の動作	27
4.9	HopAttraction() の動作	28
4.10	ユークリッド距離による力場算出方法	29
4.11	ComputeDistanceBasedForceLevel() の動作	29
4.12	SSNR と距離の関係	30
4.13	力場選択シーケンス	31
4.14	冗長問題	32
4.15	RenewNearestNodes() の動作	33
4.16	GenerateEvents() の動作	34
4.17	Poisson(lambda) の動作	34
5.1	改良版 nam	36
5.2	実装モジュール構成図	37
5.3	イベントロスト時間	41
5.4	コストパフォーマンス評価結果	42
5.5	各引力・斥力決定方式のコストパフォーマンス比較結果	44
5.6	各引力・斥力決定方式の送信 SDSP メッセージ数比較結果	45

# 表目次

1.1	Mote の仕様 . . . . .	2
2.1	センサノード単体のコスト比較 . . . . .	12
3.1	判別式と各動的拡散方式の関係 . . . . .	17
5.1	hdr_sdsp 構造体 . . . . .	38
5.2	パラメータ設定 (1) . . . . .	39
5.3	パラメータ設定 (2) . . . . .	39
5.4	パラメータ設定 (3) . . . . .	40
5.5	パラメータ設定 (4) . . . . .	40
5.6	パラメータ設定 (5) . . . . .	40
5.7	センサネットワーク vs. モバイルセンサネットワーク . . . . .	43

# 第1章

---

## 序論

---

本章では，本研究の意義および本論文の内容構成について述べる．

## 1.1 研究背景

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，計算機を集約できるようになった [20]．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[10](図 1.1)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．Mote の仕様を表 1.1 にまとめた．

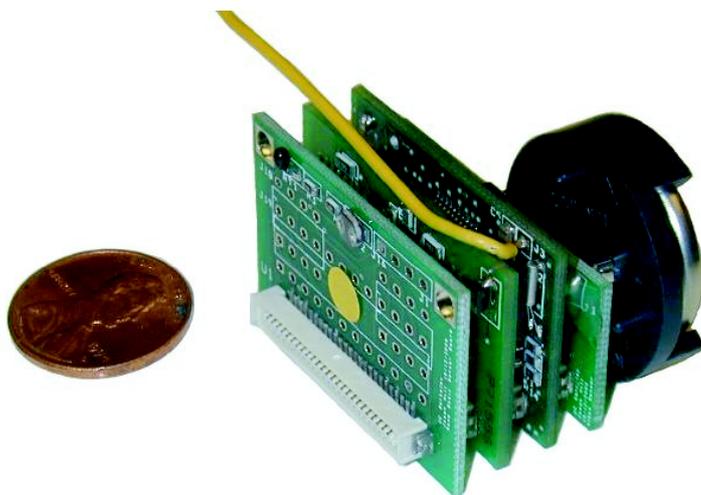


図 1.1: 無線アドホック・センサプラットフォーム：Mote

表 1.1: Mote の仕様

CPU	8bit-RISC ATmega103L (4MHz)
Flash メモリ	128K bytes
SRAM	4K bytes
EEPROM	4K bytes
無線周波数	916MHz ISM バンド
OS	TinyOS
センサ (センサボードを接続することでセンサの種類を増やせる)	音 光 温度 加速度 磁気

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 [6]，[15] 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) [17] 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 [29]，[28] 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．

## 1.2 本研究の目的

本研究の目的は，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センシング精度でデータを取得することである．この目的を達成するために，本研究はセンサノードが自律的に移動するモバイルセンサネットワークを提案し，センサノードの動的再配置を行う．これにより，センサノード散布後にその分布が偏っても，各センサノードが自律的に移動して，興味ある環境情報を発見できる．また，センシング精度を向上させる場合には，単一のセンサノードで興味ある環境情報を監視するのではなく，複数のセンサノードが同じ対象を監視・協調動作し精度向上を実現する．本論文では上記の手法を実現する SDSP 方式を設計・実装する．

## 1.3 本論文の構成

本論文は全6章より構成される．次章では，センサネットワークについて概説し，センサネットワークにおける研究課題について言及する．本論文では，センサネットワークにおける研究課題のうち，センサノードの配置問題に焦点を当て，同時にセンサノードの配置を動的に変更可能なモバイルセンサネットワークを提案する．第3章では，センサノードの動的再配置手法に関して考察する．最後に，動的再配置を実現した研究を紹介

し，本研究と比較する．第4章にてSDSP方式を設計する．SDSP方式は，変則 Random Waypoint モデルを利用して調査対象空間を走査し，調査対象が発見されるとアプリケーションによって指定されたセンシング精度に合わせて周囲のセンサノードを集める．そのアルゴリズムの詳細を説明する．第5章において本方式のシミュレーション結果をまとめる．性能評価として，コストパフォーマンス評価を行う．また，既存のセンサネットワークとの比較および考察も行う．第6章では，まとめと今後の課題について述べ，本論文を締めくくる．

## 第2章

---

# モバイルセンサネットワーク

---

本章では、既存のセンサネットワークについて概説し、本研究が注目する技術について説明する。また、センサネットワークにおける問題点としてセンサノードの配置問題について触れ、その解決アプローチとしてモバイルセンサネットワークを提案する。

## 2.1 センサネットワークの概要

センサネットワークには、センサノードとシンクノードの2種類のノードが存在する。センサノードは表 1.1 に示した機能を提供するが、中でも特徴的な機能は無線通信機能とバッテリー駆動である。各センサノードは自律的に動作し、温度、湿度、光量などを測定する。取得したデータ(センサデータ)は、センサノードの無線通信機能およびバッテリーを使用してシンクノードへ転送される。

一方、シンクノードはセンサノードよりも計算機能や計算機資源に優れている。例えばセンサノードよりも記憶媒体の容量が  $10 \sim 10^6$  倍であったり、32 ビット CPU が搭載されている。また、シンクノードはセンサノードからなる無線網との接続性に加えて、インターネットなどの有線通信基盤網との接続性も有している。ゆえに、有線基盤網上のホストはシンクノードが格納するセンサデータを取得したり、またシンクノードを通じてセンサノードにセンサデータの要求や制御命令を発行できる。シンクノードは、こうした有線基盤網上のホストに対応するため、常時動作することが求められる。したがって、シンクノードはセンサノードと異なり、太陽電池や電力線による電力確保が常に可能なノードとして動作する。

図 2.1 にセンサネットワークにおけるネットワーク構成を示す。センサネットワークのネットワーク構成はマルチホップ無線網と有線基盤網に大別できる。無線通信機能を有したセンサノードは図 2.1 のようなマルチホップアドホックネットワークを構成し、実線がマルチホップ無線網のトポロジを表している。実線で繋がれていないセンサノードは孤立ノードと呼ばれ、シンクノードへの通信経路を確保できないノードである。大量の孤立ノードの発生は、マルチホップ無線網を分断し、大規模センサネットワークの実現を妨げる。これによって、アプリケーションは、大規模な情報収集を行えなくなる。センサネットワークにおける課題のひとつはこうした孤立ノードの発生の抑制にある。

センサネットワークのネットワーク構成は、MANET(Mobile Ad-hoc Networks)[14] 分科会が提唱するアドホックネットワークによく似ているが、有線基盤網の存在を仮定している点が MANET と異なる。また、センサネットワークでは電力消費を考慮した通信が必要であるが、MANET ではセンサネットワークほど省電力通信が求められてはいない。これは、ネットワークを構成するノードの性質によるもので、アドホックネットワークがノートブック PC によって構成されるのに対し、センサネットワークでは、バッテリー容量が極端に少ない<sup>1</sup>センサノードによって構成される。

また、アプリケーションの観点から見ると、センサデータそのものに加えてセンサデータがいつ・どこで検出されたかという時刻情報および場所情報もセンサネットワークで求められる。例えば、地震を観測するアプリケーションの場合、震度だけでなくその地震が何時にどこで発生したのかも把握したい情報である。時刻情報は各センサノードにタイマ機能が備わっていれば取得可能である。しかし、場所情報に関しては大量のセンサノードを散布するため、センサノードがどこに配置されたかユーザはあらかじめ把握できない。したがって、各センサノードが自身の配置場所を特定し、センサデータにそ

<sup>1</sup>Mote が単三乾電池 2 個で動作するのに対して、ノートブック PC は乾電池より大容量のリチウム電池が使用されている。

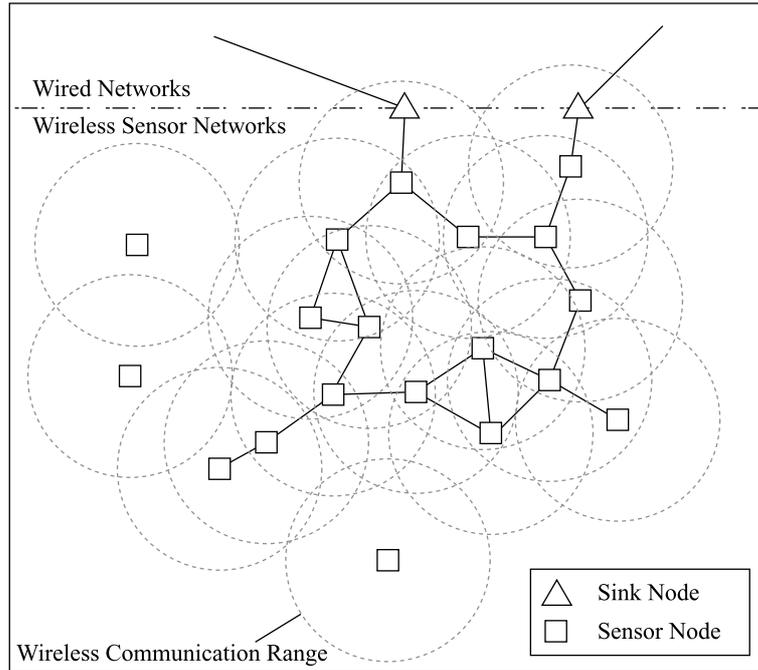


図 2.1: センサネットワーク環境

の情報を付加する必要がある。

さらに、センサノードの配置もセンサネットワークにおけるアプリケーションにとっては重要な要素である。センサノードの落下地点に偏りが生じた場合、アプリケーションの要求する場所を監視できなくなる恐れがある。また、仮にアプリケーションにとって興味ある場所を監視できたとしても、ノイズやセンサデータの不確定性によってアプリケーションが期待するセンシング精度で対象物や現象を観測できない可能性がある。したがって、センサノードをどのように配置するかは、センサネットワークにおける大きな課題である。

ここで、本研究が目指す主要技術や考慮すべき項目を下記にまとめる。

(a) 省電力通信

省電力通信の方法には主にデータリンクレイヤでの解決方法とネットワークレイヤでの解決方法がある。データリンクレイヤでの解決方法の代表例として Span[5] がある。Span は、IEEE802.11b[12] で規定されている省電力モード (power saving mode) を利用するだけでなく、ルーティングを行うノード (Coordinator) を電力消費量に併せて交替させている。Coordinator 以外のセンサノードは通信が必要ない場合に通信機能を停止させ、電力消費を抑制する。Span はデータリンクレイヤで実装されるため、あらゆるルーティングプロトコルを省電力化できる利点を持つ。一方、ネットワークレイヤでの解決方法として GAF[32] が提案されている。GAF では、調査対象空間をある一定の大きさの格子に分割する。各格子内に存在するセンサノードはルーティングを行う稼働ノードとルーティングを行わない休止ノードに分かれる。稼働ノードはある電力量を消費すると休止ノードになり、周辺のセン

サノードのバッテリー容量が少なくなると再び稼働ノードとして動作する．本研究では，こうした省電力通信を利用し，センサノード同士が協調し合うものとする．

(b) 取得したデータの場所

取得したデータの場所を得るには，センサノード自身が設置されている場所を特定しなければならない．センサノードの特定手法は，GPS(Global Positioning System)のような絶対位置が取得可能なセンサを全ノードに搭載する手法と，Localizationと呼ばれる絶対位置と相対位置の両方を利用するハイブリッド手法に分かれる．前者は，GPSを全てのノードに組み込む手間が発生するが，ほぼ正確に位置を特定できる．一方，後者の手法は全てのセンサノードにGPSを組み込むのではなく，一部のセンサノードにGPSを組み込む．他のセンサノードはGPSを搭載したセンサノードとの相対距離を超音波 [26] や無線通信デバイスの電波強度 [19] などを利用して測定し，絶対位置を算出する．本研究では，各々のセンサノードがGPSを搭載しているか，あるいはLocalization処理によって全センサノードの絶対位置を特定できることを前提とする．

(c) センサノードの配置

既存研究では，センサノードの配置に関してほとんど考慮されていない．これは，アプリケーションが要求する環境情報を必ず取得できるようなセンサノード配置が仮定されているためである．センサネットワークにおけるルーティングアルゴリズムの代表例である Directed Diffusion [13] でも，センサノードの配置は重要視されていない．また，Mainwaringらは [17] において海鳥の子育てを調査するため，実際に Berkley 大学の Mote を現地に設置している．この研究では巣穴の位置や数は別の調査によって既知であり，既知の巣穴に人が直接 Mote を設置している．このように，調査箇所が限定されており，人が直接センサノードを配置する場合にはセンサノードを適切に設置できる．本研究では，センサノードを人が直接設置するのが困難なアプリケーション，例えば遭難者探索や地雷調査などを対象にセンサノードの配置を考える．

## 2.2 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。

## 2.2.1 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、

$$D = \frac{2 \sum_{i=1}^N S_{c_i}}{\pi S_{rz}} \quad (2.1)$$

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。 $D < 1$  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。

## 2.2.2 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定性)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、遭難者探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 [4] 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.

## 2.3 研究アプローチ

本研究では, 第 2.2 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 2.1 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 [22], [21] を利用することでセンシング精度を向上できる.

本節では, モバイルセンサネットワークの概要を述べ, 既存のセンサネットワークとの比較を行う.

### 2.3.1 モバイルセンサネットワーク

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 2.2). 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 [2], [24] 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 2.2-(b) および式 2.2 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のに対して, 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 2.1  $D \geq 1$  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 2.1  $D < 1$  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.

$$R_c > R_s \quad (2.2)$$

本論文では, 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

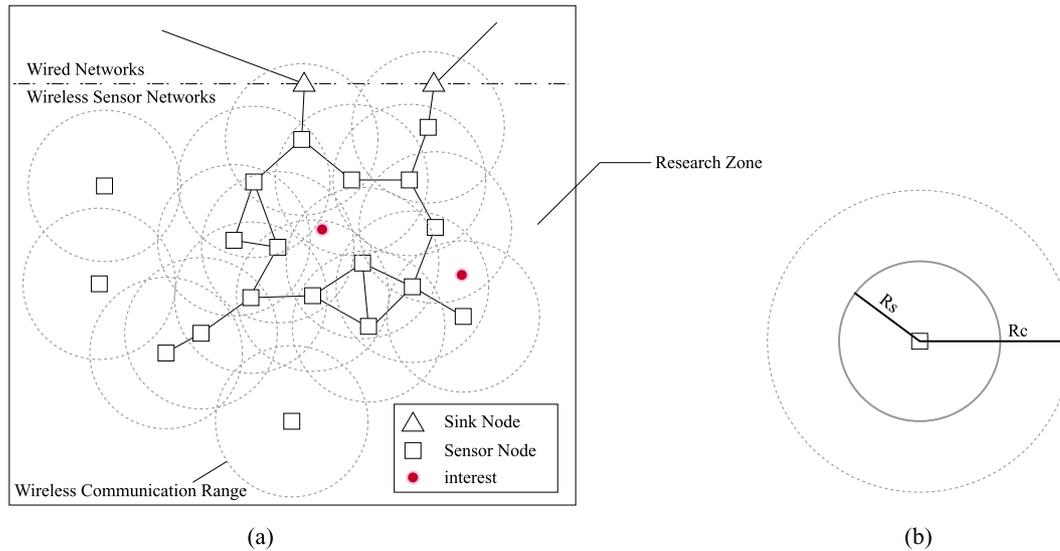


図 2.2: モバイルセンサネットワーク環境

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．本論文では 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「 $10\text{pg} - \text{TEQ}/\text{m}^3$  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．

### 2.3.2 センサネットワーク vs. モバイルセンサネットワーク

本項では，既存のセンサネットワークと本研究が提案するモバイルセンサネットワークとの比較を行う．比較項目はノードの大きさ，経済的コスト，ノードの回収率の3点である．

#### ノードの大きさ

既存のセンサネットワークにおける利点のひとつは，小型なセンサノードを実現することで一度に大量のノードを散布できる点にある．輸送能力が一定のとき，より体積の少ないセンサノードを実現すれば，それだけ散布できるセンサノード数も増加する．しか

し、逆にノード単体の大きさが増加すると、一度に散布できるノード数は減少する。センサノードに移動機能を追加することは、ノードの大きさの増大につながる。したがって、同じ条件下(同じ輸送能力)でセンサノードを散布する場合、既存のセンサネットワークより、モバイルセンサネットワークの方が散布できる量が少なくなる。しかし、モバイルセンサネットワークでは、各ノードが移動することで数の不足も補えるため散布量の減少によりモバイルセンサネットワークのセンシング性能が劣化する訳ではない。

## 経済的コスト

センサノードの機能を必要最小限に抑えることで経済的コストの削減が可能である。モバイルセンサネットワークではセンサノードは自律的に移動可能でなければならないため、既存のセンサネットワークにおけるノードと比較してコストがかかる。表 2.1 にそれぞれのノードの単価をまとめた。このことから、モバイルセンサノードの実現は、センサノードの実現より費用が3倍以上かかることがわかる。しかし、表 2.1 に示したモバイルセンサノードは、16 ビットマイクロコントローラを搭載し、記憶媒体の容量が Mote の 250 倍であることから単純な比較は行えない。

表 2.1: センサノード単体のコスト比較

センサノード例	価格(千円)	移動性
Berkley Mote	100 (材料費だけでは数千円)	無
AmigoBOT	550	有

## ノードの回収率

ノードの回収率とは、調査時に散布したセンサノード数に対する調査完了後に再び回収できたセンサノード数の比である。調査完了後にセンサノードを回収できれば、別の目的にそのセンサノードを使用できる。したがって、ノード回収率が高ければ、新たに必要となるセンサノード数が少なく済むのみでなく、その際に発生する経済的コストを低く抑えることが可能である。モバイルセンサネットワークでは、各センサノードは自律的に移動でき、調査完了後、回収地点に全センサノードを集結させられる。一方、既存のセンサネットワークでは広域にセンサノードが分散した場合、全センサノードを回収すること困難である。すなわち、モバイルセンサネットワークの方が既存のセンサネットワークよりセンサノードの有効利用を実現できる。

## 第3章

---

### モバイルセンサノードの配置

---

本章では，モバイルセンサノードの再配置を動的拡散手法と動的再配置手法の2種類に分類する．動的拡散手法は監視エリア問題を解決し，動的再配置手法は精度問題に対処する．また，既存研究を紹介し論考する．

### 3.1 動的拡散手法

動的拡散手法は、モバイルセンサノードが調査対象空間全体に広がることで監視エリア問題を解決する方法である。この動的拡散手法には、動的幾何拡散方式、動的ニューラル拡散方式および動的ランダム拡散方式の3つの方式が考えられる。本節では、それぞれの特徴および基本動作を細説する。

#### 3.1.1 動的幾何拡散方式

動的幾何拡散方式とは、調査対象空間を(正)多角形に分割し、その交点にモバイルセンサノードを配置させる手法である。図3.1-(1)にモバイルセンサノードの初期配置を示す。各センサノードには移動すべき位置があらかじめ設定されており、その位置に移動することでセンサノード分布の偏りを解消する。動的幾何配置は調査対象空間に均一にセンサノードが広がり、図3.1-(2)のように拡散する。

この手法ではセンシングエリアの重なりを減らし、効率良くかつ広域にモバイルセンサノードを拡散できるが、判別式2.1が1以上のときに特に有効である。また、移動に伴い各モバイルセンサノード同士で通信する必要がないため、移動に要する電力消費以外に余計なエネルギー損失がない。

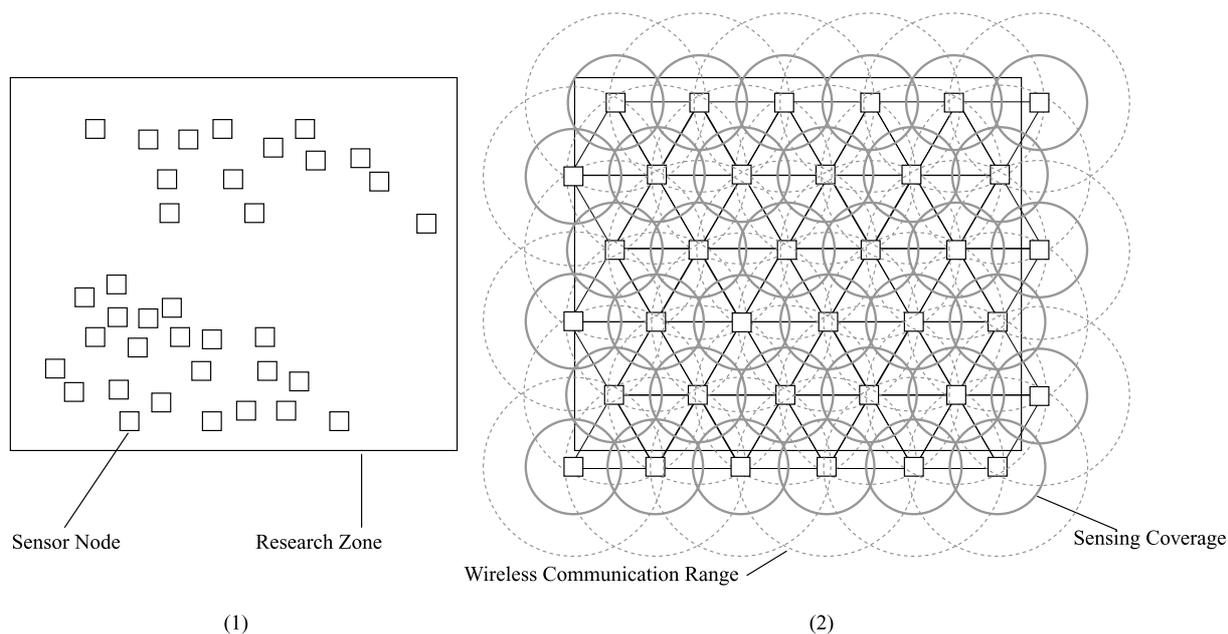


図 3.1: 動的幾何拡散方式

### 3.1.2 動的ニューラル拡散方式

動的ニューラル配置とは、ニューラルネットワークにおける教師なし学習を利用した動的再配置手法である。学習アルゴリズムには自己組織化マップ (SOM)[16] や Neural Gas(NG) アルゴリズム [18] , Growing Nueral Gas(GNG) アルゴリズム [9] などが存在する。これらアルゴリズムの基本動作は調査対象空間の任意の位置を選択し、その位置に最も近いモバイルセンサノードを抽出し、学習関数によって新しい移動位置を決定する。この学習を繰り返すことで、調査対象空間全体に広がる。

動的ニューラル配置方式の初期配置および最終形態を図 3.2 に示す。アルゴリズムによって学習関数が異なるため最終形態は多少ことなるが、動的幾何配置方式に近似した形状が得られる。

この手法では、学習過程において、リーダーとなるモバイルセンサノード(リーダーノード)と制御パケットが必要となる。リーダーノードは任意の位置を選択し、その位置に最も近いモバイルセンサノードを見つけ出すために要する。また、新しい移動位置もリーダーノードが計算し、その情報を抽出したモバイルセンサノードに通知する。

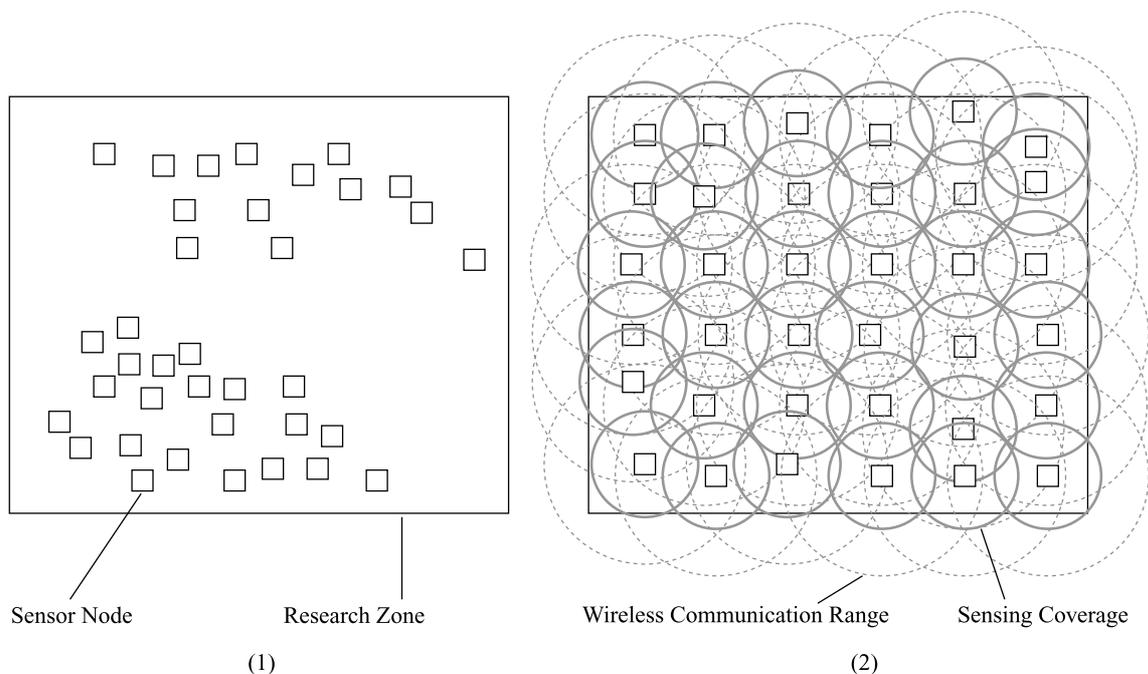


図 3.2: 動的ニューラル拡散方式

### 3.1.3 動的ランダム拡散方式

動的ランダム拡散方式は、モバイルセンサノードがランダム歩行を行うことで調査対象空間全体に広がる手法である。図 3.3 に動的ランダム拡散方式の動作例を示す。図 3.3-(1) および (2) はある瞬間のモバイルセンサノードの拡散状況を表している。この方式は先

のふたつの方式と異なり，拡散完了条件が存在しない．このため，移動による電力消費は動的幾何拡散方式や動的ニューラル拡散方式より増大してしまう．したがって，動的ランダム拡散方式には移動による電力消費を抑制したモデルが望ましい．

Random Waypoint モデル [3] は，省電力ランダム歩行を実現するモデルとして最適である．このモデルでは，ランダム歩行中に休止する時間を設け，その時間内ノードは停止する．このため，常時ランダム歩行するモデルより各モバイルセンサノードのバッテリー消費は少なく済む．Random Waypoint モデルにおいて各モバイルセンサノードは調査対象空間の中から目的地を一様分布に基づいて決定する．また，移動速度についても  $[0, v_{max}]$  の間から一様分布に基づき決定し，目的地まで等速直線運動する．目的地に到達すると一定時間休止し，その後，再び新しい移動先と速度を決定し同様の移動を繰り返す．調査対象空間の境界にモバイルセンサノードが達した場合反射し，速度は変わらず方向のみ変更し移動を続ける．

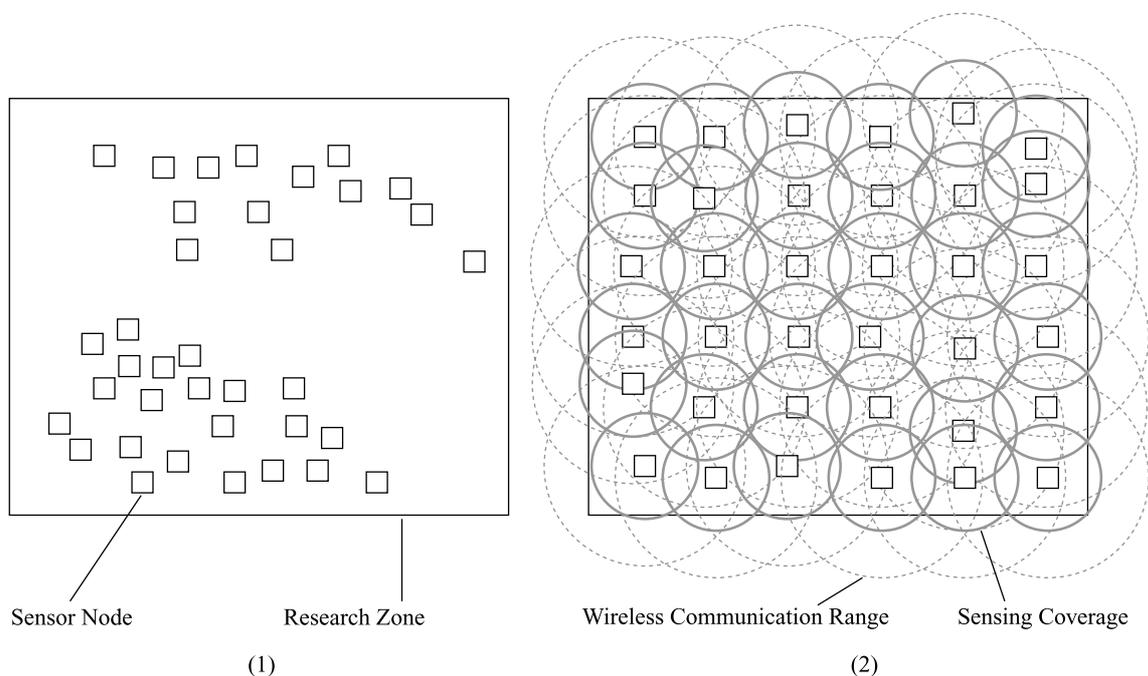


図 3.3: 動的ランダム拡散方式

### 3.1.4 各動的拡散手法の比較

本項では各動的拡散手法を耐故障性，制御パケットの量，エネルギー損失の面から比較する．またイベント発生数と各拡散方式との関係についても触れる．

耐故障性の観点から見ると，動的幾何拡散方式ではいくつかのモバイルセンサノードが故障すると，そのノードの担当観測地点はどのノードからも監視されない．また，動的ニューラル拡散方式の場合にも定常状態になった後に，いくつかのモバイルセンサノードが故障すると，そのノードの担当観測地点は監視できず，interest を見逃す可能性がある

る．一方，動的ランダム方式では幾つかのモバイルセンサノードに障害が発生しても，他のモバイルセンサノードが移動することで補完できる．したがって，耐故障性の面で評価すると，動的ランダム配置方式が優れている．

次に通信面での定性的評価を行う．動的幾何拡散方式や動的ランダム拡散方式を採用した場合，移動に伴う通信は発生しない．しかし，動的ニューラル拡散方式ではリーダ決定や移動位置伝達に各モバイルセンサノードで通信を行わなければならない．特に動的ニューラル拡散方式では，学習を行う度にリーダノードは移動位置を伝達するため，制御パケットを送信しなければならない．したがって，制御パケット送信に伴う電力消費を避けるには，動的幾何拡散方式か動的ランダム拡散方式を利用すべきである．

最後にエネルギー損失について触れる．動的ランダム拡散方式は他の二方式を異なり，収束条件(定常状態)が存在しない．つまり，アプリケーション・タスク完了まで各モバイルセンサノードは移動し続けなければならない．よって，動的ランダム拡散方式を採用した場合には，省電力な移動方法について考慮しなければならない．

ここで，各動的拡散方式と判別式 2.1 との関係を表 3.1 に示す．条件式に対して対応可能な場合は「yes」で示し，未対応あるいは対応不可能な場合には「no」で示す．表から，動的ランダム拡散方式を採用した場合，判別式のどの条件に対しても対応することが可能である．これは，各モバイルセンサノードの位置が他の二方式に比べて固定的でないためであり，汎用性が高いことを示している．

表 3.1: 判別式と各動的拡散方式の関係

動的拡散方式	$D > 1$	$D = 1$	$D < 1$
動的幾何拡散方式	yes	yes	no
動的ニューラル拡散方式	yes	no	no
動的ランダム拡散方式	yes	yes	yes

完全に調査対象空間を覆うことはできないが，それに近い状態までは広がる．

以上のことを総合的に判断して本研究では拡散方式に動的ランダム拡散方式を採用する．

### 3.2 動的再配置手法

精度問題を解決するには，2種類のアプローチが存在する．ひとつは，センサノードに搭載されているセンサ自体を高性能にすることである．この手法は，経済的コストの上昇を招くが，単体で対象物の調査を行え，必要とするセンサノード数を減らせる．もうひとつは，同一の対象物を観測するのに複数のセンサノードで行う手法である．ノイズやセンサデータの不確定性は複数のセンサノードから得られたセンサデータを統合することで解決できる．また，誤検知は複数のセンサノードによって冗長性を高めることにより解決可能である．

本研究では、2つのアプローチのうち後者を選択し精度問題に対処する。理由は前者の手法では耐故障性の面が全く考慮されていないためである。これに対して、後者の手法では観測中にいくつかのモバイルセンサノードが故障しても、周辺に存在するモバイルセンサノードを新たに呼び寄せることにより、精度問題を解決することが可能である。

動的再配置手法は、同一対象物を監視するモバイルセンサノードの配置状況を動的に変更させる方法であり、本研究で選択したアプローチに基づいている。動的再配置手法では、同一対象物を監視するモバイルセンサノードが現状の数では精度問題を解決できない場合に、周囲に存在する他のモバイルセンサノードに支援を求める。また、逆に精度問題を解決するうえですでに十分なモバイルセンサノードが存在する場合には幾つかのモバイルセンサノードをその対象物調査から解放する。ここで重要となるのが、同一対象物を監視するモバイルセンサノードの数をいかに調整するかであり、この手法の課題である。

### 3.3 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 [31], [11], [23] を参照する。

#### 3.3.1 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 [31]。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ト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。[31] 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。

### 3.3.2 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 [11]。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な行われぬ。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。

### 3.3.3 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 [23]。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。

## 第4章

---

### SDSP の設計

---

本章では，自律型動的配置方式 SDSP 方式 (**Self-organizing Dynamic Sensor Placement**) の概要とそのアルゴリズムの設計について述べる．本アルゴリズムでは，各モバイルセンサノードが三種類のモードを切り替えることで監視エリア問題，精度問題に対応する．まずはじめに SDSP の基本動作を説明し，次にアルゴリズムの詳細を述べる．

## 4.1 SDSPの基本動作

本節ではSDSP方式の基本動作を詳述する．図4.1は本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* , *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 *SENSE\_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(本論文では，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．詳細は第4.2節)を受信すると，*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．

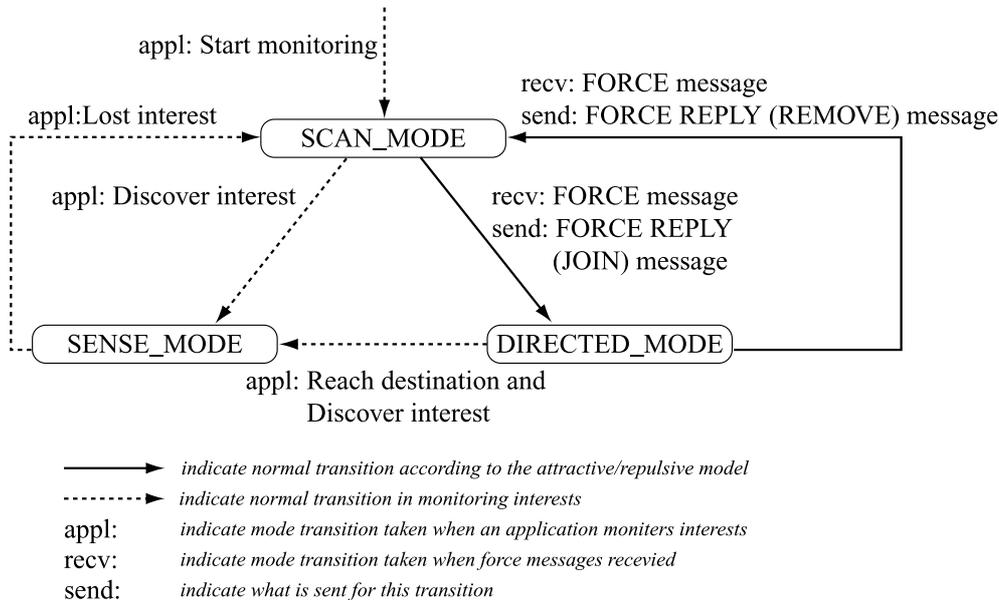


図 4.1: 状態遷移図

### 4.1.1 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．

### 4.1.2 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する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後述する引力・斥力モデルを導入する．

### 4.1.3 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．

### 4.1.4 引力・斥力モデルの導入

本方式において，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 4.2 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，センス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 4.2 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．

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<sup>1</sup>．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．

---

<sup>1</sup> 引力・斥力がゼロであることを意味する．

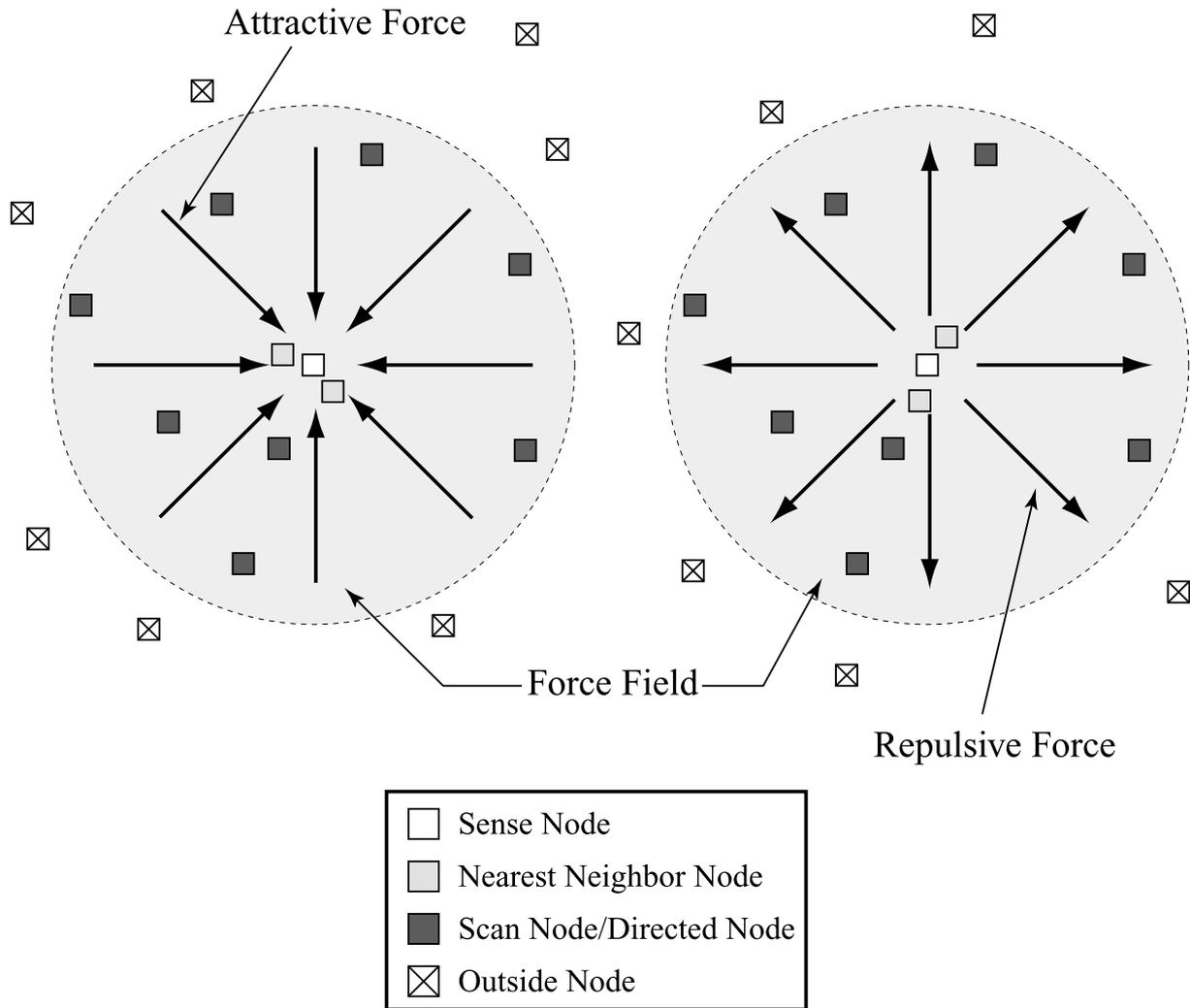


図 4.2: 引力・斥力モデルの概念図

## 4.2 引力・斥力制御方式の設計

引力・斥力制御方式の基本動作を図 4.3 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。本論文では、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ブノードはランダム時間待機後に 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、そのメッセージを発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、それを発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。

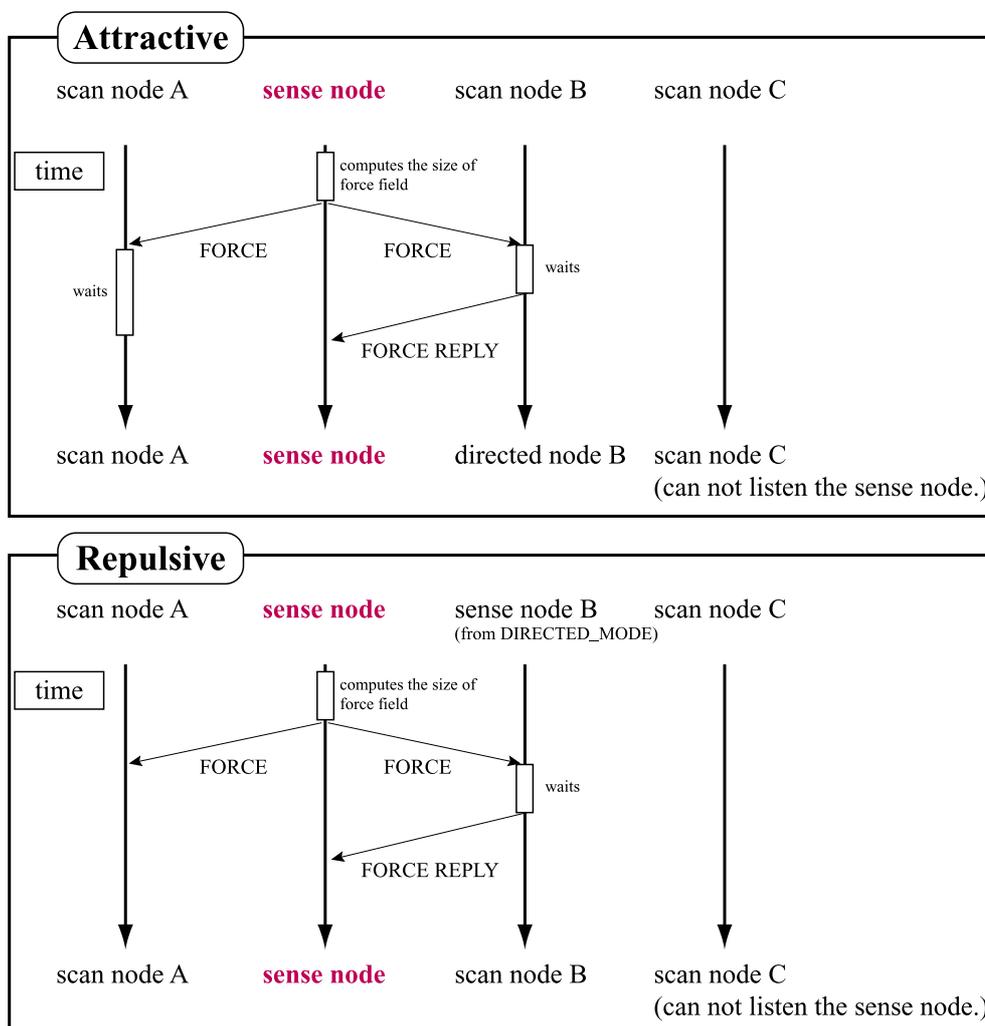


図 4.3: 引力・斥力制御方式の基本動作

本節では、まず SDSP メッセージの形式について設計する。SDSP 方式において、各モバイルセンサノードはこの SDSP メッセージを通じて情報のやりとりを行う。次に引力・斥力決定方法について説明を加える。また、FORCE メッセージ内に書き込まれる移動位置情報の決定方法について考察する。最後に、同一 interest に複数のセンスノードが集まってしまう冗長問題について論考する。

### 4.2.1 SDSP メッセージ

SDSP メッセージは、FORCE メッセージ、FORCE REPLY メッセージ、HELLO メッセージ、HELLO REPLY メッセージの 4 種類に大別できる。図 4.4 に FORCE メッセー

ジおよび FORCE REPLY メッセージの構造を示す。FORCE メッセージは、引力・斥力情報を載せたメッセージであり、周辺のモバイルセンサノードの挙動に影響を与える。LEVEL フィールドは、引力・斥力の大きさを表し、正の値の場合は引力、負の値の場合は斥力とする。0 の場合には引力・斥力ともに存在しないことを意味し、このメッセージを受信したノードは何も応答を返さない。また、この値が MAX\_LEVEL である場合は、そのメッセージを発したセンサノードおよび最近傍ノードを強制的に SCAN\_MODE に遷移させる。

一方、FORCE REPLY メッセージは、FORCE メッセージに対する応答メッセージであり、引力メッセージに対する返答を返す場合は JOIN フラグをセットし、斥力メッセージに対する応答の場合には REMOVE フラグをセットする。

#### FORCE message

TYPE	TTL	HOP COUNT	LEVEL	ORIG_ADDR	DST_ADDR	ORIG_LOC	DST_LOC
[FORCE]	このメッセージの有効ホップ数	このメッセージを発したノードからのホップ数	引力・斥力算出方法に基づいた値	このメッセージを発したノードの MAC アドレス	ブロードキャストアドレス	このメッセージを発したノードの位置情報	このメッセージを受信したノードが移動すべき位置情報

#### FORCE REPLY message

TYPE	FLAG	TTL	HOP COUNT	ORIG_ADDR	DST_ADDR
[FORCE]	JOIN (引力メッセージへの応答) or REMOVE (それ以外)	このメッセージの有効ホップ数	このメッセージを発したノードからのホップ数	このメッセージを発したノードの MAC アドレス	FORCE メッセージを発したノードの MAC アドレス

図 4.4: FORCE メッセージ / FORCE REPLY メッセージ

図 4.5 に HELLO メッセージおよび HELLO REPLY メッセージを示す。これらのメッセージは、HELLO メッセージを発したモバイルセンサノードの無線通信可能範囲に何台の他のモバイルセンサノードが存在するか調べるのに用いられる。HELLO メッセージは、無線通信可能範囲 1 ホップだけに送信すればよいので TTL フィールドの値は 1 に設定される。このメッセージを受信した他のモバイルセンサノードは、HELLO REPLY メッセージを生成する。HELLO REPLY メッセージでは、REPLY フラグをセットし、HELLO メッセージを発したモバイルセンサノード宛にメッセージを送り返す。

### 4.2.2 引力・斥力決定方法

本項では三種類の引力・斥力決定方法について言及する。力場を定める方法には、メッセージの (1) ホップ数を利用する方法と、(2) 距離情報を利用する方法、さらに (3) 無線通信デバイスの受信電波強度を利用する方法がある。以下でこれらについて詳説する。

### HELLO message

TYPE	TTL	HOP COUNT	ORIG_ADDR	DST_ADDR
[HELLO]	1	このメッセージを發したノードからのホップ数	このメッセージを發したノードのMACアドレス	ブロードキャストアドレス

### HELLO REPLY message

TYPE	FLAG	TTL	HOP COUNT	ORIG_ADDR	DST_ADDR
[HELLO]	REPLY flag	1	このメッセージを發したノードからのホップ数	このメッセージを發したノードのMACアドレス	FORCEメッセージを發したノードのMACアドレス

図 4.5: HELLO メッセージ / HELLO REPLY メッセージ

#### (1) ホップ数による算出

本算出方法では、力場の有効範囲はホップ数によって決定する。図 4.6-(a) では、1 ホップの場合の力場を定義している。ホップ数が 2 の場合の力場は図 4.6-(b) のように表現できる。

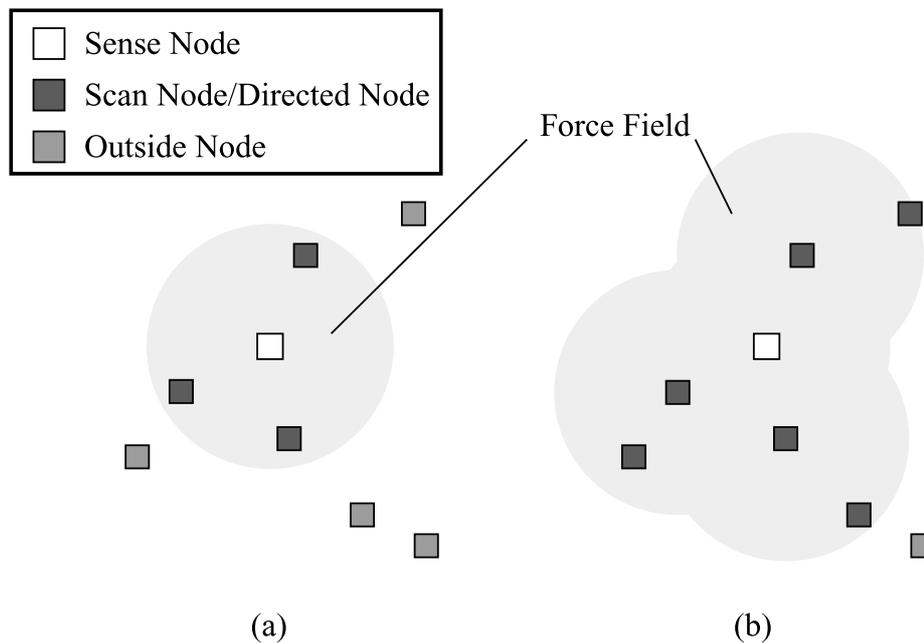


図 4.6: ホップ数による力場の有効範囲設定

ホップ数による手法を用いた場合、力場は無線通信範囲  $R_c$  の大きさに依存する。一般に同じ 1 ホップであっても  $R_c$  が大きい場合には、 $R_c$  が小さい場合に比べてより多くのモバイルセンサノードにメッセージが行き渡る可能性がある。この  $R_c$  の大きさを考

慮した力場決定を行うには，センスノードはまず1ホップの場合に何台のモバイルセンサノードにメッセージが受信されるか測定し，その情報に基づいて力場を決定しなければならない．

これらを踏まえた引力・斥力算出アルゴリズム COMPUTEHOPBASEDFORCELEVEL() は，図 4.7 のようになる．

```

COMPUTEHOPBASEDFORCELEVEL()
  if COUNTNEARESTNEIGHBORNODES() > required_node_num then
    level ← HOPREPULSION()
  else if COUNTNEARESTNEIGHBORNODES() < required_node_num then
    level ← HOPATTRACTION()
  else
    level ← 0
  return level

```

図 4.7: ComputeHopBasedForceLevel() の動作

```

HOPREPULSION()
  level ← -1
  return level

```

図 4.8: HopRepulsion() の動作

手続き COUNTNEARESTNEIGHBORNODES() によって最近傍ノード状況を確認し，*required\_node\_num* との比較計算を行う．*required\_node\_num* 変数は，アプリケーションによって指定される整数値(ただし，非負)であり，センシング精度を向上させるために必要なモバイルセンサノード数を意味している．最近傍ノード数が既にこの数値を上回っている場合には，新たなモバイルセンサノードを必要とせず，手続き HOPREPLUSION()(図 4.8) によって斥力の強さを決定する．本アルゴリズムにおいて引力の大きさを常に  $-1$  に設定している理由は，1 ホップ以内のモバイルセンサノード，特に最近傍ノードに対して斥力が働けばよいためである．

逆に，COUNTNEARESTNEIGHBORNODES() の値が変数 *required\_node\_num* を下回った場合には，手続き HOPATTRACTION() (図 4.9) を実行し，引力を決定する．

手続き HOPATTRACTION() 内では，COUNTONEHOPNODES() が呼ばれ，センスノードから1ホップ以内に存在する他のモバイルセンサノード数を調べる．この手続きにおいてセンスノードはHELLOメッセージを送信し，一定時間それに対するHELLO REPLYメッセージを回収する．この情報をもとに，引力の大きさを決定するのだが，本アルゴリズムでは1ホップあたり  $n$  台のモバイルセンサノードが存在すると見積り， $n \times \sum_{k=0}^{l-1} n^k$

```

HOPATTRACTION()
  n ← COUNTONEHOPNODES()
  if n ≥ required_node_num then
    level ← 1
  else
    l ← 2
    while n × ∑k=0l-1 nk < required_node_num
      and l < MAX_HOP
      l ← l + 1
    level ← l
  return level

```

図 4.9: HopAttraction() の動作

によってホップ数を増やした場合に FORCE メッセージを受信できるノード数を計算する。この値が *required\_node\_num* を超えるか、ホップ数が *MAX\_HOP* を上回ると引力の大きさが決定される。

本手法では、引力・斥力計算時に HELLO メッセージとその HELLO REPLY メッセージを必要とし、他の 2 方式と比べ余計な電力消費を伴う。

## (2) ユークリッド距離による算出

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 4.10-(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 4.10-(b))。

本算出方法では、力場の有効範囲(有効距離)を  $\infty$  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 4.11 のようになる。

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ればよい。

$$L_{own} - L_s \leq LEVEL \quad (4.1)$$

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 メッセージを発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

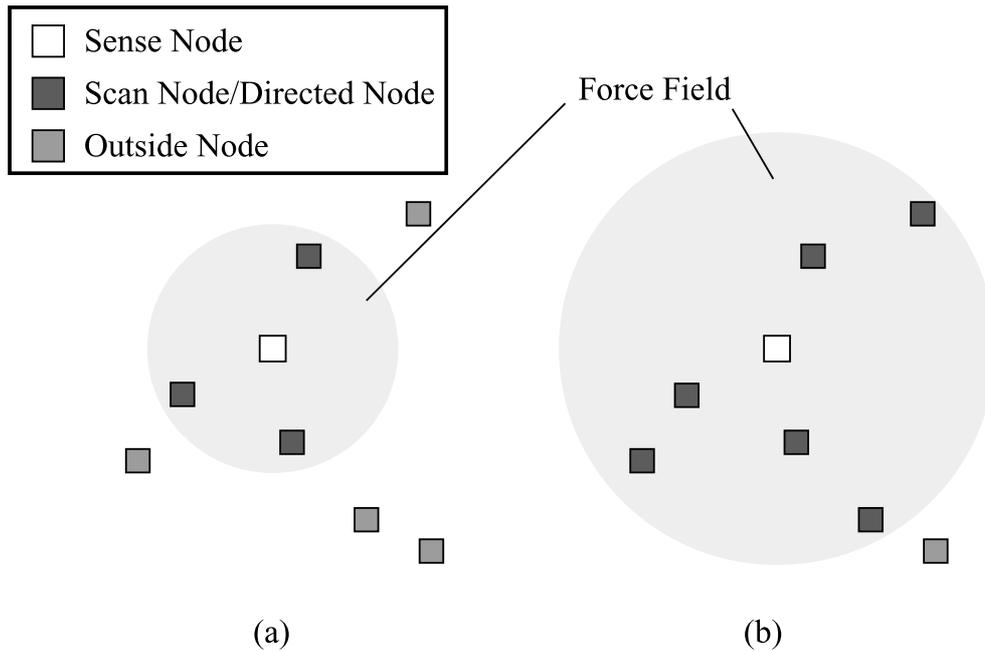


図 4.10: ユークリッド距離による力場算出方法

```

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 4.11: ComputeDistanceBasedForceLevel() の動作

士の減算結果は、2点間のユークリッド距離と定義する。

### (3) 受信電波強度による算出

(2)の方法では、距離によって力場有効範囲を制御した。本手法でも受信電波強度によって力場有効範囲を制限する。無線電波強度は、通信する2点間の距離に応じて変化することが知られている。したがって、電波強度を利用することで、距離による力場の制御と同様のことが行える。

図4.12は、SSNR (Smoothed Signal to Noise Ratio) と距離との関係を示している。IEEE802.11b 準拠のデバイスによって取得された電波強度と数式 4.2 を利用すると SSNR 値が得られる。

$$ssnr = (1 - \alpha) \times old\_ssnr + \alpha \times cur\_ssnr \quad (4.2)$$

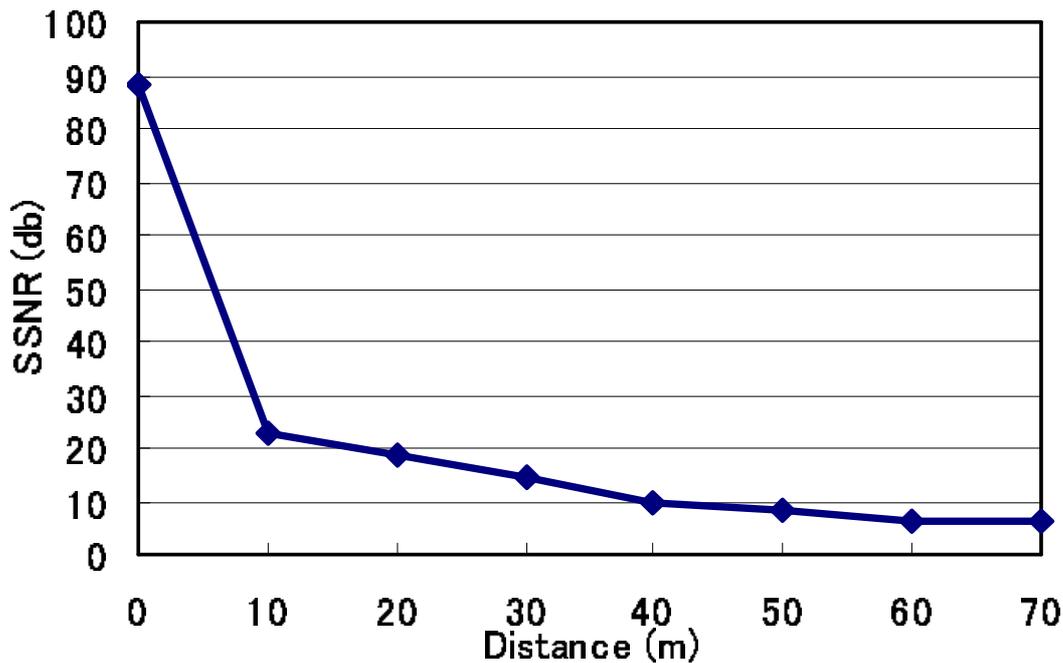


図 4.12: SSNR と距離の関係

$\alpha$  は重みであり， $[0,1]$  の範囲の値である． $cur\_ssnr$  とはパケットの受信によって得られた SNR 値であり， $old\_ssnr$  は前回の計算によって得られた SSNR 値である．

この SSNR 値によって力場有効範囲を決定することが可能であるが，受信電波強度は環境の変化を受けやすく，障害物がある場合には必ずしも図 4.12 のような曲線を描かない．したがって，本研究では本方式を利用した力場決定および引力・斥力決定は行わない．

### 4.2.3 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ードの最近傍として登録される（ディレクティッ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，ディレクティッ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，ディレクティッ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いはない．最初の移動位置を受信した FORCE メッセージ内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 .

#### 4.2.4 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。

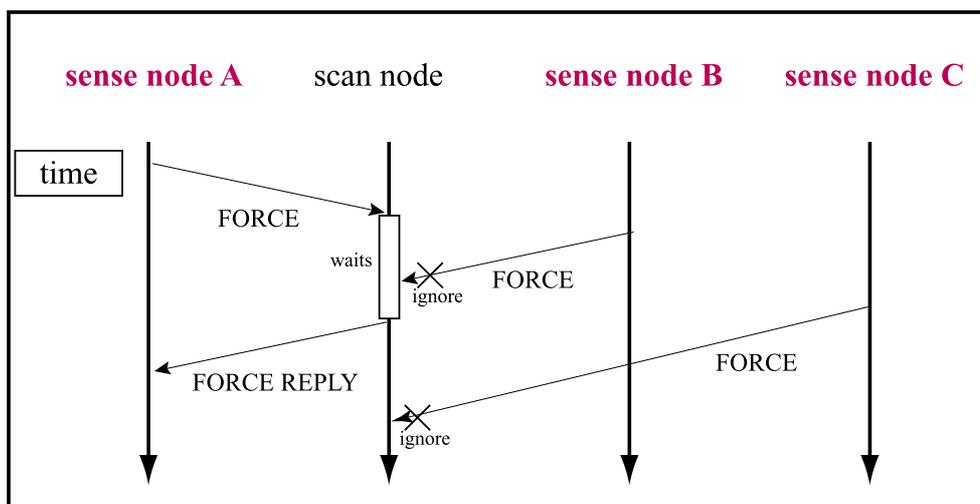


図 4.13: 力場選択シーケンス

図 4.13 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。

## 4.2.5 冗長問題

図 4.14 のように複数のモバイルセンサノードが同じ interest を発見して集中してしまう可能性がある(図 4.14-(a))。この場合、各モバイルセンサノードは個々でセンシング精度達成のための必要台数を確保してしまい、過剰にモバイルセンサノードが集まってしまふ(図 4.14-(b))。本論文では、これを冗長問題と呼び、以下でその解決方法を示す。本論文では冗長問題に対する解決手法として適者生存方式と協調動作方式を考察する。

各センサノードが同一 interest を監視しているかの判断には、センサノード  $i$  の位置情報  $L_i$  と自身の位置情報  $L_{own}$  を利用し、その距離関係が  $L_i - L_{own} \leq 2 \times R_s$  ( $R_s$  はセンシング範囲) である場合に共通の interest を検知しているものとする。

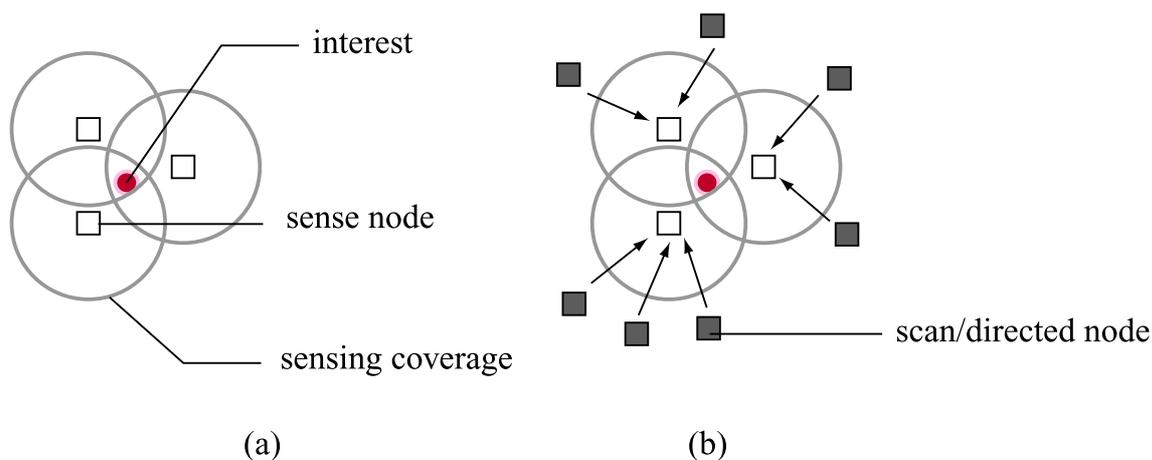


図 4.14: 冗長問題

### 適者生存方式

適者生存方式は、同一 interest を監視する複数のセンサノードをただひとつのセンサノードに絞り込む方式である。本方式は、能動的適者生存方式と受動的適者生存方式の二種類の実現方法が存在する。

- 能動的適者生存方式  
能動的適者生存方式では、各センサノードがランダムナンバーを含んだメッセージを通信し合い、一番大きな数のセンサノードが勝者としてその場所に留まる。しかし、本方式では最近傍ノードに関する情報のやりとりのために FORCE メッセージ以外に別のメッセージを常に送受信しなければならない。これはモバイルセンサノードの通信による電力消費を増加させることになる。したがって、本研究ではこの方式を採用しない。
- 受動的適者生存方式  
受動的適者生存方式では、各センサノードが送信する FORCE メッセージを受信し、

メッセージ内の LEVEL フィールドを抽出し、LEVEL の値が 0 に近いものを勝者とする。この方式では、FORCE メッセージのみで適者を選択でき、他の手法と比べモバイルセンサノードの通信に余計な負荷が加わらない。したがって、本研究ではこの方式を利用して冗長問題に対処する。

### 協調動作方式

協調動作方式では、各センサノードが互いに自身の位置情報および最近傍ノード数を通知しあい、その情報をもとに引力や斥力さらに力場を計算する。

ここで、最近傍ノード集合を  $G$  とし、センサノード  $i$  における  $G$  を  $G_i$  と表記し、自身の最近傍ノード集合は  $G_{own}$  とする。各センサノードは、引力・斥力決定時に、手続き RENEWNEARESTNODES() (図 4.15) を実行し、同一 interest を観測している全センサノードの最近傍ノード集合を回収する。その情報をもとに引力・斥力を再計算する。

```

RENEWNEARESTNODES()
  repeat until listening period ends
     $G_{own} \leftarrow G_{own} \cup G_i$ 
  return  $G_{own}$ 

```

図 4.15: RenewNearestNodes() の動作

しかし、本方式では最近傍ノードに関する情報のやりとりのために FORCE メッセージ以外に別のメッセージを常に送受信しなければならない。これはモバイルセンサノードの通信による電力消費を増加させることになる。したがって、本研究ではこの方式を採用しない。

## 4.3 イベント発生モデル

本節ではイベント発生モデルの設計を行う。イベント発生モデルの最も一般的なものはポアソン分布に基づいたイベント発生パターンである。例えば地雷をある地域に均一にばら撒いたときにその地雷に遭遇する場合などがポアソン分布に基づいたイベント発生モデルである。

ここで特筆すべきは、本研究が提案する SDSP 方式は特定のイベント発生モデルに特化した方式ではないことである。したがって、ポアソン分布以外のイベント発生モデルにも対処することは可能であるが、本論文ではポアソン分布に基づいたイベント発生モデルを用い、シミュレーション評価を行う。

イベント発生アルゴリズムは手続き GENERATEEVENTS() (図 4.16) によって実現される。平均イベント発生数を  $\lambda$  とし、ポアソン分布に基づいて  $n$  個のイベントが新たに発生する。新しいイベントが発生した場合には、手続き NEWEVENT() を呼び出し、イベン

トの発生時間，イベント TTL，発生場所を決定する．イベント TTL とは，そのイベントが持続する時間を意味し，その時間が過ぎると消滅する．またイベント発生場所は，一様分布に基づいて決定される．

```
GENERATEEVENTS()  
   $n \leftarrow \text{POISSON}(\textit{lambda})$   
   $i \leftarrow 0$   
  while  $i < n$   
     $\textit{new} \leftarrow \text{NEWEVENT}()$   
     $\textit{EVENTS} \leftarrow \textit{EVENTS} \cup \textit{new}$   
     $i \leftarrow i + 1$ 
```

図 4.16: GenerateEvents() の動作

手続き GENERATEEVENTS() で使用される POISSON(*lambda*) は図 4.17 のようになる．ただし，EXPONENTIAL(*x*) は自然対数の底 *e* の *x* 乗を計算する手続きであり，UNIFORM() は [0,1] の範囲で一様分布に基づいて実数値を抽出する手続きとする．

```
POISSON(lambda)  
   $n \leftarrow 0$   
   $y \leftarrow \text{EXPONENTIAL}(\textit{lambda}) \times \text{UNIFORM}()$   
  while  $y > 1$   
     $y \leftarrow y \times \text{UNIFORM}()$   
     $n \leftarrow n + 1$   
  return  $n$ 
```

図 4.17: Poisson(*lambda*) の動作

## 第5章

---

### シミュレーション

---

本章では，SDSP方式の性能評価を行う．シミュレータに **Network Simulator version 2 (ns-2)** を使用し，Agent モジュールとして SDSP 方式を実装した．シミュレーション評価では，コストパフォーマンス評価，既存のセンサネットワークとの比較および考察，さらに引力・斥力制御方式の比較検討を行う．

## 5.1 シミュレーションモジュールの実装

本論文の提案する SDSP 方式の評価をシミュレーション上で行う。シミュレーションには、ns-2 (Network Simulator version 2)[8] を使用し、シミュレーション過程の視覚化には nam (Network Animator)[7] を改良して表示させる (図 5.1)。

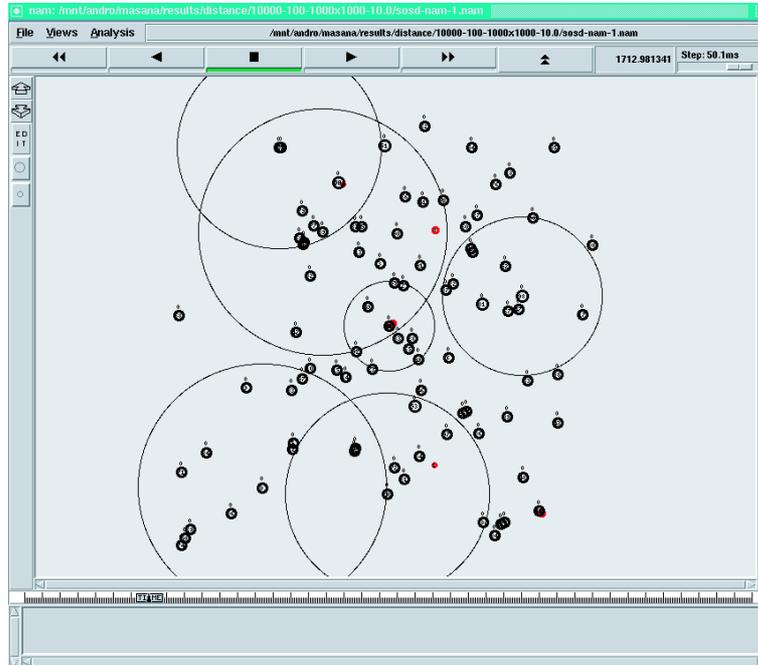


図 5.1: 改良版 nam

SDSP 方式を実現する主要なシミュレーションモジュールは、図 5.2 のように構成される。ns-2 が提供する Agent モジュールを継承した SdspAgent モジュールは、SDSP メッセージを作成し、interest に集合するモバイルセンサノード数を調整する。通常、Agent モジュールはトランスポート層に位置するが、本モジュールはリンクレイヤ層の上位層として実現される。SdspAgent モジュールをトランスポート層として実現すると MAC 層に対する制御を行えず、1 ホップ内に存在するノードにのみ SDSP メッセージを送信あるいは受信させることが困難となるためである。したがって、SdspAgent はリンクレイヤ層へのポインタ `ll_` を有し、リンクレイヤ層のモジュールに対して直接メッセージ送信命令を発行できる。

また、ns-2 が提供する MobileNode モジュールを継承した MobileSensorNode モジュールを新たに実装し、変則 Random Waypoint 移動やセンサ機能を実現した。センサ機能では、イベントを発見し、その時刻、場所 (モバイルセンサノードの存在位置)、観測したイベントの値、さらにそのイベントを見失った時刻を保持する。また本モジュールは、センシング範囲やセンサのサンプリングレート、interest 情報 (反応する値)、センシング精度向上に必要なモバイルセンサノードの台数を指定でき、この情報に基づいて interest 探索を行う。

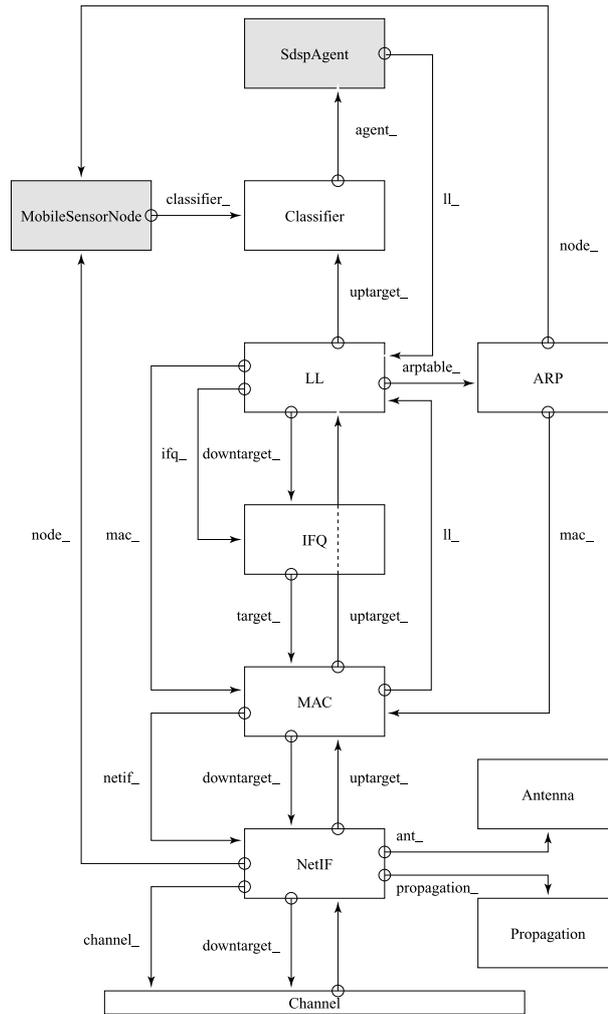


図 5.2: 実装モジュール構成図

SDSP 方式における通信で利用されるメッセージ形式の一部を表 5.1 に示す。構造体に存在する `ns_loc_t` は、モバイルセンサノードの位置情報を表す構造体であり、シミュレーションフィールド内の X 座標、Y 座標、Z 座標情報を含む。

## 5.2 シミュレーションモデル

シミュレーションでのフィールドサイズおよびシミュレーション時間を表 5.2 のように定めた。各モバイルセンサノードはフィールドサイズ情報を保持し、調査対象空間外のフィールドに移動しないように動作する。本実装において、各モバイルセンサノードは、境界に達すると反射する。

また、モバイルセンサネットワーク環境ではノード数を 100 台、150 台、200 台に定め、移動速度を変えて評価を行った。一方、センサネットワーク環境ではノード数を 600 台に定めてシミュレーションを行った。

表 5.1: hdr\_sdsp 構造体

```

struct hdr_sdsp{
    char type_;           // SDSP message type.
    char flag_;          // for future use
    char ttl_;           // Hop limits.
    char hop_;           // The number of hops from
                        // the originator to
                        // the destination node.

    int level_;          // force level that you
                        // can change.
    int saddr_;          // The node address of the node
                        // which originated the SDSP
                        // message.
    int daddr_;          // The node address of the
                        // destination node.

    ns_loc_t src_loc_;   // The originator's location
    ns_loc_t dst_loc_;   // The destination location where
                        // the receiver node should go

    snipped...
};

```

また、シミュレーションで用いる (1) 無線モデル、(2) センサモデル、(3) イベント発生モデルを下記のように定める。

### 5.2.1 無線モデル

無線モデルでは、無線信号の伝搬状況を設定する。本シミュレーションにおいて、無線伝搬モデルは Two-Ray Ground Reflection モデル [25] を用いる。このモデルは、直接波と地面からの反射波を考慮したモデルであり、より現実的な無線通信環境をシミュレートする。伝搬距離  $d$  での受信出力  $P_r$  は次式により予測される。

$$P_r(d) = \frac{P_t G_t G_r h_t^2 h_r^2}{d^4 L} \quad (5.1)$$

$P_t$  は送信出力、 $G_t$ 、 $G_r$  はそれぞれ送信局のアンテナ利得、受信局のアンテナ利得である。 $h_t$  と  $h_r$  は送信局側と受信局側のアンテナの地面からの長さを表す。また、 $L$  はシステムのロスファクターである。これらの値を表 5.3 のように定める。ns-2 では、無線周波数帯として 914MHz 帯を使用している。

表 5.2: 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 <sup>1 2</sup> , 20m/sec <sup>1</sup> , 30m/sec <sup>1</sup>
センサネットワーク	
ノード数	600 台

<sup>1</sup> 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

<sup>2</sup> 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

表 5.3: パラメータ設定 (2)

パラメータ	設定値
送信出力 $P_t$	0.2818W
アンテナの利得 $G_t, G_r$	1.0
アンテナの高さ $h_t, h_r$	1.5m
ロスファクター $L$	1.0
無線周波数 $f$	914MHz

加えて、MAC層でのパラメータ設定を表 5.4 にまとめた。パラメータ  $\gamma_1, \gamma_2, \gamma_3$  は搬送波検知可能な出力最小値，受信可能な出力最小値，受信可能な信号対干渉雑音比の最小値である。受信可能出力最小値  $\gamma_2$  を  $1.42681 \times 10^{-8}$  に定めることにより，無線通信可能範囲の半径  $R_c$  を約 100m に設定できる。IEEE802.11b 準拠の無線通信デバイスでは，通信可能範囲の半径は通常見通し 100m であり，この値はシミュレーションにおいて妥当であると言える。

## 5.2.2 センサモデル

センサモデルでは，センサの設定値を定める。本シミュレーションではセンサの設定値としてセンシング範囲  $R_s$ ，サンプリングレート  $S_{rate}$ ，センシング精度達成必要台数  $n$  を決定する。これらの値を表 5.5 に示す。センシング範囲やサンプリングレートは実際に存在する人感センサの値を考慮して設定した。したがって，本パラメータ設定は現実的な値であると言える。

表 5.4: パラメータ設定 (3)

パラメータ	設定値
搬送波検知出力 $\gamma_1$	$1.559 \times 10^{-11}W$
受信出力 $\gamma_2$	$1.42681 \times 10^{-8}W$
信号対干渉雑音比 $\gamma_3$	10.0dB

表 5.5: パラメータ設定 (4)

パラメータ	設定値
センシング範囲 $R_s$	10m
サンプリングレート $S_{rete}$	40Hz
必要台数 (自身を除いて) $n$	4台

### 5.2.3 イベント発生モデル

シミュレーションにおいて、イベント発生はポアソン分布に従って決定される。イベント発生における諸パラメータを表 5.6 のように定めた。これらのパラメータによって平均同時イベント出現数は、 $\frac{\lambda \times TTL}{T}$  によって求められ、平均総イベント発生数は  $\frac{\lambda \times ST}{T}$  のようになる。ここで  $ST$  はシミュレーション時間とする。本パラメータ設定を適用した場合における平均同時イベント出現数は 5 個/msec になり、平均総イベント発生数は 36 個となる。本パラメータ設定は、[17] における生態調査よりさらに厳しい条件を設定している。したがって、本シミュレーションでは、モバイルセンサネットワーク (SDSP 方式) が苛酷な条件下でも機能するか検証可能である。

表 5.6: パラメータ設定 (5)

パラメータ	設定値
平均イベント発生確率 (1msec あたり) $\lambda$	$5.0 \times 10^{-6}$
イベント単位時間 $T$	$1.0 \times 10^{-3}sec$
イベント TTL $TTL$	1000sec

## 5.3 シミュレーション評価

本シミュレーション評価では、以下の 3 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

評価 3 引力・斥力決定方式の比較

### 5.3.1 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  $E_i$  は次式を利用することで計算可能である。

$$E_i = \alpha m_i v_i t_i \tag{5.2}$$

$\alpha$  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  $\alpha m_i$  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 5.3 に定めて計算できる。

$$E_i = v_i t_i \tag{5.3}$$

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 5.3 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成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。

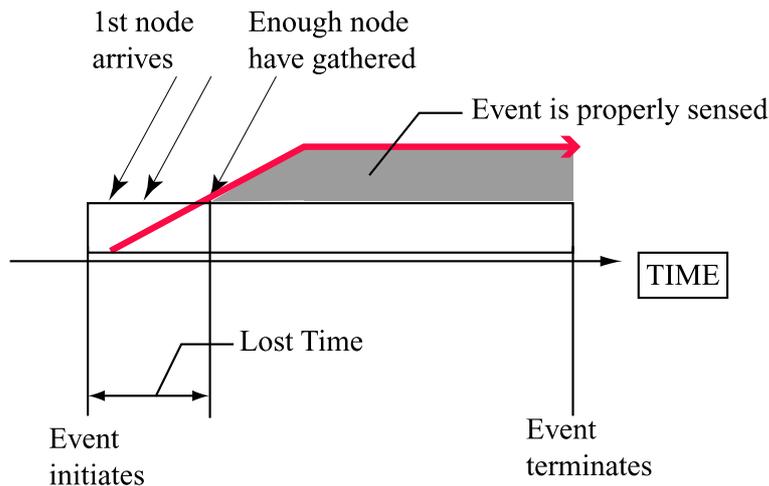


図 5.3: イベントロスト時間

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5.4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。

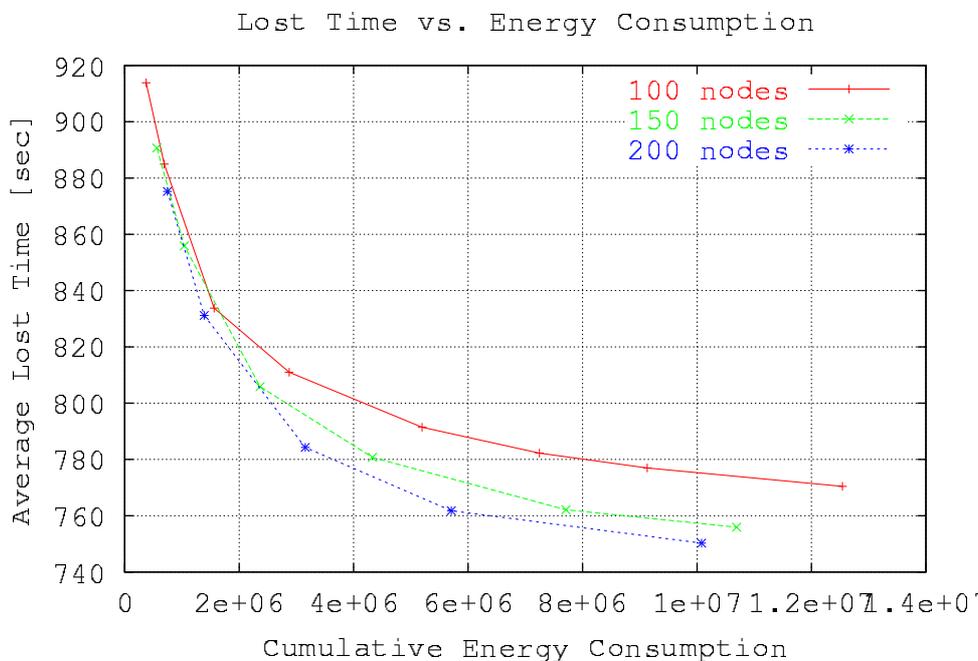


図 5.4: コストパフォーマンス評価結果

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ともよい(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．

### 5.3.2 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，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．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 5.7 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．

また，第 2.3.2 項において示した経済的コストを考慮すると，経済的コストに対するパフォーマンスはモバイルセンサネットワークの方が優位である．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．

表 5.7: 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)

### 5.3.3 評価 3 : 引力・斥力決定方式の比較

本項では、TTL による引力・斥力決定方式と距離による引力・斥力決定方式の特質をシミュレーションによって明らかにする。本シミュレーション評価では、まず各方式のコストパフォーマンス評価を行う。コストパフォーマンス評価では評価 1 と同様に、累計エネルギー消費と平均イベントロスト時間との関係を調べる。次に、平均イベントロスト時間とシミュレーション時間内に送信された総 SDSP メッセージ数から、各方式を評価する。

本シミュレーション評価ではモバイルセンサノード数を 100 台とし、モバイルセンサノードの移動速度を  $0.5m/sec$ 、 $1m/sec$ 、 $2.5m/sec$ 、 $5m/sec$ 、 $10m/sec$ 、 $15m/sec$ 、 $20m/sec$ 、 $30m/sec$  と値を変化させた。

#### シミュレーション結果とその考察

図 5.5 に各方式の移動による累積エネルギー損失と平均イベントロスト時間の関係を示す。本グラフより、モバイルセンサネットワーク全体に与えられるエネルギー量が増加するにつれて、TTL による方式の方が距離による方式より平均イベントロスト時間を減らすことが可能であるが、両者の差はほとんどない。

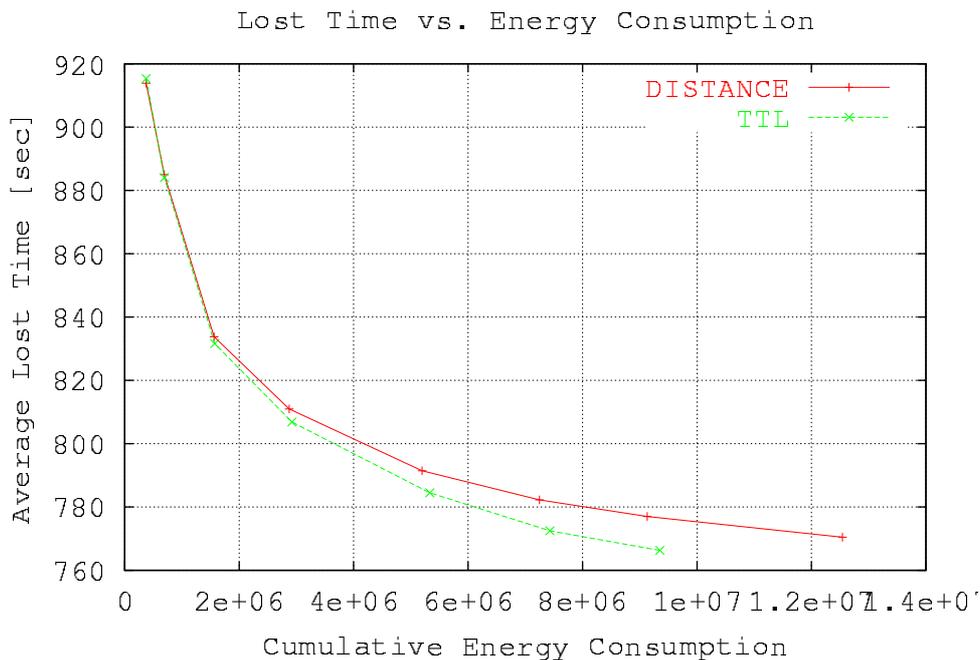


図 5.5: 各引力・斥力決定方式のコストパフォーマンス比較結果

次に、平均イベントロスト時間とシミュレーション時間内に送信された SDSP メッセージ数の関係を図 5.6 とに示す。両グラフとも、平均イベントロスト時間の値が少なくなるにつれて、総送信 SDSP メッセージ数は増加する。この現象は次のことから説明できる。

多くのモバイルセンサノードが interest を発見でき，かつアプリケーションが指定したセンシング精度を得るのに十分な数のモバイルセンサノードが集まれば平均イベントロスト時間は減少する．すなわち，平均イベントロスト時間の低下はセンスノードの増加を意味し，センスノードの増加によって総送信 SDSP メッセージ数は増加することになる．

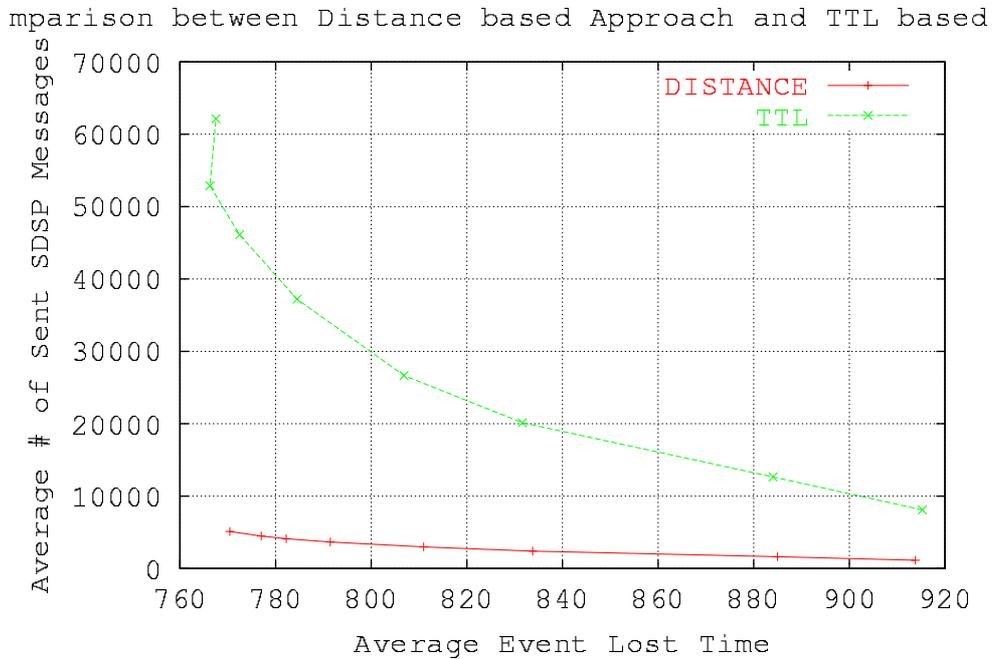


図 5.6: 各引力・斥力決定方式の送信 SDSP メッセージ数比較結果

ここで，距離による方式と TTL による方式とを比較する．距離による引力・斥力決定方式を採用した場合，TTL による方式より最低 1/2 以下のメッセージ数で済む．これは距離による引力・斥力決定方式ではセンスノードは FORCE メッセージしか送信しないためである．一方の TTL による引力・斥力決定方式では，FORCE メッセージに加えて HELLO メッセージも送信しなければならない．また，TTL による算出方法では，力場有効範囲を広げると TTL が増加し，その分 FORCE メッセージを距離による方式より多く送信しなければならない．

したがって，コストパフォーマンス評価および総 SDSP メッセージ数評価から引力・斥力決定方式には距離による算出方式を採用すべきである．

## 第6章

---

### 結論

---

本章では、今後の課題および展望について触れ、本研究についてまとめる。

## 6.1 今後の課題

今後の課題に関しては主に次のふたつについて考慮しなければならない。

### 実機での検証

本論文では、自律的に移動するセンサノードを仮定してシミュレーションを行った。しかし、シミュレーション環境では、無線の反射波による影響やノイズに対しての検証を行っていない。また、障害物が存在した場合の各センサノードの動作に関する考慮されていない。したがって、実機 (LinuxBot[30], AmigoBot[1] 等) を用いて本アルゴリズムが正しく動作するか検証しなければならない。同時に、通信によって消費される電力および移動によって消費される電力を計測し、省電力で機能するかの実験も必要だろう。

### ターゲットトラッキングへの対応

本論文でのイベント発生モデルは、イベントの位置は固定で発生するモデルであった。しかし、今後はイベント自体が移動するモデルも考慮しなければならない。例えば、小動物を追跡するようなアプリケーションを想定した場合、イベントは固定ではなく、その位置は時間によって変化する。また、引力・斥力制御方式も移動イベントに対応できるように改良を加える必要があるだろう。特に、周辺のモバイルセンサノードを集結させる際の効率的な移動位置決定がターゲットトラッキングの分野における重要な課題である。

## 6.2 まとめ

本論文は、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、本論文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。

# 謝辞

本論文の執筆にあたり、御指導頂いた慶應義塾大学環境情報学部教授の徳田 英幸博士を始め、本論文の副査としてご助言頂いた慶應義塾大学環境情報学部教授の清木 康博士、同大学同学部助教授の楠本 博之博士に深謝する。

特に、慶應義塾大学大学院政策・メディア研究科助教授の西尾 信彦博士には研究の議論のために時間を割いて頂いた。また、慶應義塾大学大学院政策・メディア研究科博士課程の永田 智大氏、同大学院同研究科修士課程の堀江 裕隆氏には絶えざる励ましと御指導を賜わった。ここに記して、感謝の意を表す。

2003年1月14日  
村瀬 正名

## 参考文献

- [1] ACTIVMEDIA ROBOTICS. Amigobot. <http://www.amigobot.com/>.
- [2] BROCH, J., JOHNSON, D., AND MALTZ, D. The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks, 2002.
- [3] BROCH, J., MALTZ, D. A., JOHNSON, D. B., HU, Y.-C., AND JETCHEVA, J. A performance comparison of multi-hop wireless ad hoc network routing protocols. In *International Conference on Mobile Computing and Networking* (1998), pp. 85–97.
- [4] 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.
- [5] CHEN, B., JAMIESON, K., BALAKRISHNAN, H., AND MORRIS, R. Span: An Energy-Efficient Coordination Algorithm for Topology Maintenance in Ad Hoc Wireless Networks. In *ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom)* (2001), pp. 85–96.
- [6] 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.
- [7] ESTRIN, D., HANDLEY, M., HEIDEMANN, J., MCCANNE, S., XU, Y., AND YU, H. Network Visualization with the VINT Network Animator Nam. Technical Report 99-703b, University of Southern California, March 1999.
- [8] FALL, K., AND VARADHAN, K. *ns Manual*. The VINT project, UC Berkeley, LBL, USC/IDI, URL: <http://www.isi.edu/nsnam/ns/>, May 2001. Work in progress.
- [9] FRITZKE, B. A Growing Neural Gas Network Learns Topologies. In *Advances in Neural Information Processing Systems 7* (1995), MIT press, pp. 625–632.
- [10] HILL, J., AND CULLER, D. A wireless embedded sensor architecture for system-level optimization. In *UC Berkeley Technical Report* (2002).

- [11] 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).
- [12] IEEE. Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications, 1999. IEEE Standard 802.11.
- [13] INTANAGONWIWAT, C., GOVINDAN, R., AND ESTRIN, D. Directed Diffusion: A Scalable and Robust Communication Paradigm for Sensor Networks. In *ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom)* (2000), pp. 56–67.
- [14] INTERNET ENGINEERING TASK FORCE (IETF). Mobile ad-hoc networks (MANET) working group charter, 1999. <http://www.ietf.org/html.charters/manet-charter.html>.
- [15] 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.
- [16] KOHONEN, T. Self-organized formation of topologically correct feature maps. In *Biological Cybernetics* 43 (1982), pp. 59–69.
- [17] MAINWARING, A., POLASTRE, J., SZEWCZYK, R., CULLER, D., AND ANDERSON, J. Wireless Sensor Networks for Habitat Monitoring. In *International Workshop on Wireless Sensor Networks and Appliances* (2002), pp. 88–97.
- [18] MARTINETZ, T. M., AND SHULTEN, K. J. A "neural-gas" network learns topologies. In *Artificial Neural Networks* (1991), T. Kohonen, K. Mäkiö, O. Simula, and J. Kangas, Eds., pp. 397–402.
- [19] MEGUERDICHIAN, S., SLIJEPCEVIC, S., KARAYAN, V., AND POTKONJAK, M. Localized Algorithms in Wireless Ad-Hoc Networks: Location Discovery and Sensor Exposure. In *ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHOC)* (2001), pp. 106–116.
- [20] MEMS EXCHANGE. MEMS Clearinghouse. <http://www.memsnet.org>.
- [21] 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.
- [22] 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.

- [23] PARKER, L. E. Cooperative Robotics for Multi-Target Observation. *Intelligent Automation and Soft Computing* 5, 1 (1999), 5–19.
- [24] PERKINS, C., ROYER, E., AND DAS, S. Ad Hoc On Demand Distance Vector (AODV) Routing, 2002.
- [25] RAPPAPORT, T. S. *Wireless Communications, Principles and Practice*. Printice Hall, 1996.
- [26] SAVVIDES, A., HAN, C.-C., AND STRIVASTAVA, M. B. Dynamic Fine-Grained Localization in Ad-Hoc Networks of Sensors. In *ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom)* (2001), pp. 166–179.
- [27] SILBEY, G. T., RAHIMI, M. H., AND SUKHATME, G. S. Robomote: A Tiny Mobile Robot Platform for Large-Scale Ad-hoc Sensor Networks. In *International Conference on Robotics and Automation* (2002), vol. 2, pp. 1143–1148.
- [28] WANT, R., HOPPER, A., FALCAO, V., AND GIBBONS, J. The Active Badge Location System. *ACM Transaction on Information Systems* 10, 1 (January 1992), 91–102.
- [29] WARD, A., JONES, A., AND HOPPER, A. A New Location Technique for the Active Office, 1997.
- [30] WINFIELD, A. Linuxbot. <http://www.ias.uwe.ac.uk/~a-winfie/linuxbot/linuxbot.htm>.
- [31] 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.
- [32] XU, Y., HEIDEMANN, J., AND ESTRIN, D. Geography-informed Energy Conservation for Ad Hoc Routing. In *ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom)* (2001), pp. 70–84.