

# 移動型センサネットワークにおける複数ノードのための経路探索手法

中宮 正樹 岸野 泰恵 寺田 努 西尾章治郎

大阪大学大学院情報科学研究科 〒565-0871 大阪府吹田市山田丘 1-5

E-mail: {nakamiya.masaki,yasue,tsutomu,nishio}@ist.osaka-u.ac.jp

あらまし 筆者らはこれまで、従来の移動型センサネットワークの研究では扱われていなかった実用上の問題を解決するために、コストマップを用いた移動型センサノードのための経路探索手法を提案した。提案手法により、センサのセンシング範囲やノード移動時の障害物、ノードの移動特性を考慮したノードの経路探索が可能となった。本稿では新たに、複数ノード同士の衝突を考慮したアルゴリズムと、センシング領域を指定するための問合せ言語を提案する。これらの手法により、ユーザは複数のノード用いて複数のセンシング領域を効率よくセンシングできるようになる。さらに、使用するノードの移動特性や搭載されているセンサの種類や数、センシング範囲に関する知識がなくても、問合せ言語を記述することで複数のノードがセンシング要求に応じた最適な場所へ自動的に配置されるようになる。本稿では、複数ノード同士の衝突を回避するアルゴリズムの有効性を確認するためにシミュレーションによる評価を行った。

キーワード センサネットワーク, 移動型センサノード, 経路探索, 消費電力, 問合せ言語

## A Route Planning Method for Multi Mobile Sensor Nodes

Masaki NAKAMIYA, Yasue KISHINO, Tsutomu TERADA, and Shojiro NISHIO

Graduate School of Information Science and Technology, Osaka University Yamadaoka 1-5, Suita-shi,  
Osaka, 565-0871 Japan

E-mail: {nakamiya.masaki,yasue,tsutomu,nishio}@ist.osaka-u.ac.jp

**Abstract** We have proposed a route planning method for mobile sensor nodes using *cost map*. The proposed method achieves a novel path planning that can solve several practical problems in previous works: the limitations of sensing areas, barricades on nodes' path, and restrictions on nodes' movements. In this research, we propose a route planning method that can avoid collisions among multiple nodes. Moreover, we also propose a query definition language to specify required sensing data. The proposed method can achieve effective allocation of multiple sensor nodes without knowledge about the characteristics of each node. Furthermore, we show the effectiveness of the proposed method through software simulations.

**Key words** Sensor Network, Mobile Sensor Nodes, Route Planning, Energy Consumption, Query Language

### 1. はじめに

近年のネットワーク技術や無線通信技術の発展、センサデバイス的小型化および高性能化により、センサデバイス自体に無線通信機能をもたせたセンサノードによってネットワークが形成される、センサネットワークの研究が盛んに行われている[10]。最近では、センサノードにアクチュエータを搭載し、自由に移動できるようにした移動型センサノードに関する研究が注目されている[6]。ノードに移動性をもたせることで、センシングデータ要求の動的な変化に対応でき、人が近づけない危険地帯や人の手が届きにくい場所などへのノード配置も可能となる。

筆者らはこれまでに、移動型センサノードを用いた従来研究ではあまり考慮されていなかった実用上起こりうる問題を解決するために、コストマップを用いた移動型センサノードの経路探索手法を提案した[4]。提案手法により、センサの種類によるセンシング可能な範囲の違い、ノード移動時の障害物、およびノードの移動特性を考慮し、ノードが移動に要する消費電力が最小となる経路を探索することが可能となった。

しかし、これまでの手法では単一のノードのみを想定していたため、複数のノードを同時に使用する際に起こりうるノード同士の衝突を考慮していなかった。そこで本稿ではまず、複数ノード同士の衝突を考慮した経路探索アルゴリズムを提案する。これにより、複数ノードを複数箇所のセンシング領域へ効率よ

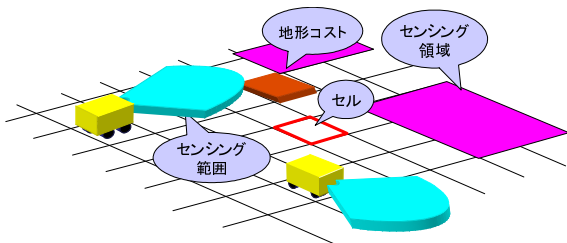


図 1 フィールドマップ

く同時に配置することが可能となる。また、従来手法では、移動型ノードを用いてセンシングを行うために、ユーザはノードに搭載されているセンサの種類やセンサデータの必要な時刻などを考慮して、どのノードをどのセンシング領域へ移動させるのかを指定しなければならなかった。そこで、移動型ノードを用いたセンサネットワークのための問合せ言語を提案する。これによりユーザは、センシングしたい場所や時刻、センサの種類といったセンシング条件を容易に設定でき、さらに使用するノードの数や移動特性、搭載されているセンサの種類や特徴に関する詳細な知識がなくても複数ノードをセンシングの目的に応じて効率的に配置できるようになる。

本研究では、ノード同士の衝突を考慮したアルゴリズムに関して、配置されるまでにノードが移動に要する電力量および全ノードの配置が完了するまでの消費時間の比較を行い、それぞれが他のアルゴリズムと比べて有効となる状況を示した。

以下、2章で想定する環境について述べ、3章で提案手法について説明する。4章でシミュレーションによる性能評価について述べ、5章で考察を行い、6章で結論と今後の課題を述べる。

## 2. 想定環境

本研究では、ビル内や地下鉄の構内などの地形が変化しない静的な環境におけるセンシングを想定し、地形の情報は図1に示すような2次元格子状に区切られたフィールドマップとしてあらかじめ与えられているものとする。フィールドマップの1つの格子を本研究ではセルと呼び、セル毎に、凹凸の多い地面や障害物といったノードの移動に影響を及ぼす情報が、地形コストとして設定されている。このような状況において、ノードの移動を制御するサーバに対して、ユーザから単一ノードではまかなえないような複数箇所での広範囲に渡るセンシングデータの要求があり、要求に対して多種のノードを複数台使用して協調的にセンシングを行う。なお、移動型ノードを用いたセンサネットワークではノードが移動する際の消費電力を低く抑えることが重要であることから、本研究ではノードの消費電力をコストとし、コストの削減を目的とする。

移動型センサネットワークで使用するノードの種類としては、車型ノードや2足歩行型ノードなど様々な種類が考えられる。それぞれ移動方法に特性をもち有利な環境が異なるため、移動型センサネットワークでは同じ種類のノードを複数利用するのではなく、様々な種類のノードを併用してセンシングすることが望ましい。また、最近は様々なセンサが日常生活で利用されており、センサネットワークにおいても、ユーザが知りたい情

表 1 想定するノード

ノード	種類	搭載するセンサ
1	車	温度、湿度
2	キャタピラ	温度、赤外線
3	車	湿度、赤外線

表 2 ノードとセンシング領域の組合せ

パターン	ノード 1	ノード 2	ノード 3
1	温度	湿度	赤外線
2	湿度	赤外線	温度

報により使用するセンサが異なるため、ノードに多種のセンサを搭載することが望ましい。このような観点から、本研究では多種のセンサを複数搭載した、多種のノードを併用したセンシングを行う環境を想定する。さらに、ノードはスタート地点からセンシング領域へ移動後は継続的にセンシングし続け、他のセンシング領域へは移動しないものとする。

## 3. 提案手法

移動型ノードを用いたセンサネットワークの利用目的は様々であり、ユーザの目的に応じて要求されるセンサデータが異なる。要求されるセンサデータには、センシングしたい範囲やセンサの種類、センシングする時刻やセンシングの精度など複数の条件が同時に設定されると考えられる。ユーザがセンシングの目的に応じて移動型ノードを利用するためには、これらの条件が容易に指定できる必要がある。

また、本研究で想定するように、各ノードが複数のセンサを搭載し様々な種類のノードを併用して複数箇所のセンシングを行う場合、ノードの配置パターンが複数存在する。例えば、温度、湿度、赤外線の3種類のセンシング要求に対し、表1に示すような2種類のセンサを搭載した3台のノードを用いてセンシングする場合、ノードの配置パターンの組合せは表2に示す2通りが考えられる。そのため、ユーザからの要求を元に各ノードの移動経路探索を行った上で合計コストが最小となる配置パターンを決定しなければならない。そこで本稿では、これらの要件を満たす、複数ノードを用いたセンサネットワークのための問合せ言語とノード同士の衝突を考慮した経路探索アルゴリズムを提案する。

### 3.1 問合せ言語

SQLを元に移動型センサネットワーク向けの問合せ言語を提案する。ユーザは必要なセンサデータの条件として、センシング領域、センサの種類、制限時間、精度を指定する。センシング領域とは、ユーザがセンサデータを取得したい場所であり、長方形で表現する。センサの種類とは、ユーザが欲しいセンサデータの種類を表し、同じ種類のセンサを搭載したノードがセンシング領域へ配置される。制限時間は、ユーザがセンサデータを取得したい時間を意味し、本稿ではノードがこの時間内に到達すればユーザはデータを取得できるものとしている。精度は、低く設定するとセンシング領域からやや離れた位置からでもセンシングし、高く設定するとセンシング領域に近づいてセ

ンシングする。

以下に問合せの構文を示す。

```
SELECT type, accuracy
FROM (x, y), (width, height)
LIMIT time
```

まず、SELECT 文でセンサの種類 *type* と精度 *accuracy* を指定する。*type* では、*temp* で温度センサ、*humid* で湿度センサといったセンサの種類を指定する。*accuracy* ではセンサのセンシング範囲がセンシング領域を完全に包含するかないかといった形で指定することができる。具体的には *accuracy* は数値で指定し、1 であればノードはセンサのセンシング範囲がセンシング領域を包含する配置場所、0 であればセンサのセンシング範囲が一部センシング領域と被る配置場所の中から、移動コストが最小のものがそれぞれ選択される。次に、FROM 文でセンシングしたい範囲を座標で指定する。*(x, y)* でセンシング領域の最左上座標を指定し、*width* でセンシング領域の横のセル数、*height* でセンシング領域の縦のセル数を指定する。さらに、LIMIT 文でセンシング領域毎の制限時間 *time* を指定でき、ノードは制限時間内にセンシング領域へ配置される。なお *time* は、指定がなければ無制限とする。

### 3.2 衝突を考慮した経路探索アルゴリズム

問合せ言語を用いてユーザが記述したセンサデータの要求を元に、衝突を考慮して複数ノードをセンシング領域へ配置させる。

#### ノードの配置場所の決定

ユーザが指定したセンサデータの種類とノードに搭載されているセンサの種類を比較し、それらが一致するセンシング領域とノードの組合せを全て列挙する。次に、1 つのセンシング領域に関して、ノードが搭載するセンサのセンシング範囲とセンシング領域が重なるようなノードの配置位置および向きを、全て列挙する。センシング領域毎にスタート地点から候補の配置位置までの経路探索を従来の経路探索アルゴリズムを用いて行い、コストが最小となる 1 つの経路を決定する。この際、ノード同士の衝突を考慮するためにノードがセルを通過する時間も同時に計算する。続いて、センシング領域とノードの全ての組合せについて経路探索を行う。ここで、ノードがセルを通過する時間を比較してノード同士の衝突を検出する。ノードが 2 セル移動する時間内に、周囲 1 セル内に複数のノードが存在する状態を衝突とする。ノードにはあらかじめ優先度がつけられており、衝突を回避するために優先度がより低いノードについて提案手法による経路の再探索を行う。最終的に、合計コストが最も小さくなる組合せを選択する。

#### 衝突を考慮した経路探索アルゴリズム

提案手法は、我々がこれまでに提案したコストマップを用いた経路探索アルゴリズムを元に行っている。まず従来手法について簡単に説明し、提案手法について従来手法との違いを述べながら説明する。従来手法では、経路探索を行う前にノードが地形の影響がない状況で周辺のセルへ移動するのに要するベース

3	2	2	2	3
5	2	1	2	5
5	6	1	6	5
5	2	1	2	5
3	2	2	2	3

(a) ベースコスト  
(ノード 90° 方向)

6	4	2	2	2
4	3	1	1	2
2	1	1	1	2
2	1	1	3	4
2	2	2	4	6

(b) ベースコスト  
(ノード 45° 方向)

図 2 車型ベースコスト

コスト (図 2) をあらかじめ測定する。測定は、移動後のノードの向きによるコストの違いを考慮して 8 方向別に行う。なお、図 2 では説明の簡単化のため向きは考慮していない。セル毎に以下の簡単な計算により *Score* を求め、コストマップを作成することでコストが最小となる経路の探索を行う。

$$Score = C + H$$

*C* は、スタート地点から注目するセルまでの移動に要する総移動コスト、*H* は、注目するセルからゴール地点まで地形コストがないと仮定した場合の移動コストである。*Score* はノードの向きにより 8 方向で区別し、1 つのセルに対し 8 方向別に *Score* を計算する。*Score* を計算する際には、ノードの向きによって図 2(a) に示す 90° 方向と図 2(b) に示す 45° 方向のベースコストを使いわけ、ノードの向きに合わせて計算する。

また、経路探索の過程ではセルは以下の 3 つの状態に分類される。

未計算： *Score* の計算が行われていないセル

OPEN： *Score* の計算が行われたセル

CLOSE： *Score* の計算が行われ、そのセルまでの最短経路が確定しているセル

最初に、スタート地点を中心としたベースコストの範囲内に含まれる全てのセルについて *Score* の計算を行い、これらのセルを OPEN 状態とする。次に、OPEN 状態のセルの中で最小 *Score* をもつセルを CLOSE 状態にし、このセルを中心としたベースコストの範囲内に含まれる全てのセルについて *Score* を計算する。以降、ゴール地点が CLOSE 状態となるまで計算を繰り返すことで経路探索を行う。また、各セルには自身の *Score* が計算される直前に注目されていたセルの位置と、そのセルからの移動経路が記録されているため、ゴール地点からスタート地点までをたどると最小コスト経路が得られる。

本稿で提案する、衝突を考慮した経路探索アルゴリズムは従来手法を元に行っている。従来手法では各セル毎に *Score* が最小となる経路の情報のみを保持していたが、提案手法では *Score* の小さい上位複数の経路情報を同時に保持する。また、*Score* の計算と同時にスタート地点からの移動に要する時間も計算し、*Time* と表記する。

ここで、図 3 に示すフィールドマップと、図 2 に示すベースコスト、および図 4 を用いて、複数ノードの衝突を考慮した経路探索アルゴリズムについて説明する。なお、この例では説明の簡単化のため移動後のノードの向きによるコストの違いを考慮しない。また、優先度の最も高いノードの経路探索が既に終了しているものとする。

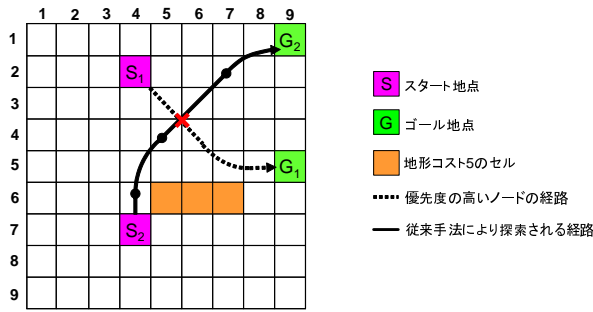


図3 フィールドマップ

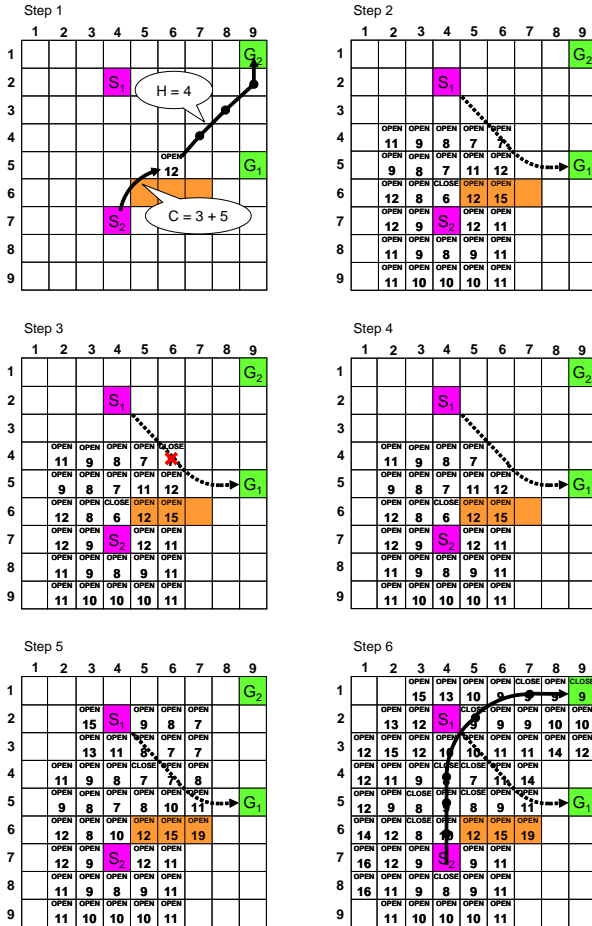


図4 経路探索アルゴリズム適用例

**Step 1** スタート地点から、ベースコストの範囲に含まれるセルの *Score* および *Time* を計算して OPEN 状態にする。スタート地点でのノードの向きが上向きであるため *Score* の計算にはベースコスト  $A$  を用いる。例えば、セル (6, 5) では  $C$  はベースコストによるコスト 3 に途中通過する地形コスト 5 を加えた 8 となり、 $H$  は 4 となるため、*Score* は 12 となる。

**Step 2** OPEN 状態のセルの中で *Score* が最小となるセル (4, 6) に注目し、CLOSE 状態にする。注目したセル (4, 6) から、ベースコストの範囲に含まれるセルの *Score* を計算して OPEN 状態にする。この時 OPEN 状態にされたセルには、直前に注目されていたセルの位置と、そのセルからの移動経路および *Time* を記録しておく。ここで、セル (2, 8), (3, 7), (6, 8) ではセル (4, 6) から計算された *Score* の方が小さいため、

*Score* と移動経路および *Time* を更新する。なお、セル (4, 6) から計算される以前の経路情報は *Score* 順に保持しておく。

**Step 3** 以降、同様の手順を繰り返すと、セル (6, 4) のように優先度の高いノードが通過するセルに注目する場合がある。ここで、記録されている通過時間と経路探索中のノードがそのセルを通過する時間 *Time* を比較し、時間差がノードが 2 セル分の移動に要する時間以下であれば衝突するものとみなす。衝突がないと判断した場合は Step 5 に進む。

**Step 4** 衝突するとみなされた場合、それまで探索してきた経路は使用できないため他の経路を探索する。この際、通過時間が異なる経路が探索できるようにするためセル (4, 6) の状態を OPEN 状態とし、CLOSE 状態の *Score* の次に *Score* が小さい経路情報をそのセルの新しい情報として更新する。この図において、セル (4, 6) ではまだ他の経路から計算されていないため、未計算状態となる。

**Step 5** Step 4 で再び OPEN 状態となったセルも含めた OPEN 状態の全セルの中で *Score* が最小となるセル (5, 4) に注目して CLOSE 状態にし、ベースコストの範囲に含まれるセルの *Score* を計算して OPEN 状態にする。

**Step 6** ゴール地点が注目されて CLOSE 状態となるまで *Score* の計算を繰り返し行う。セルには、自身の *Score* が計算される直前に注目されていたセルの位置と、そのセルからの移動経路が記録されているため、ゴール地点からスタート地点までを辿ることで、ノード同士の衝突を回避する経路が得られる。途中、優先度の高いノードの経路と交わっているが、そのセルを通過する時間に差があるため衝突しない。

#### 4. 性能評価

提案するアルゴリズムの性能を評価するためのシミュレーションを行った。地形コストは、壁などのノードが通過できない障害物を想定し、コストが無限大のもののみを配置した。ノードは、前進、後退、旋回を行う戦車型ノードと、前輪で角度をつけて前進、後退を行う車型ノードの 2 種類を考える。実際に LEGO 社 MindStorm [11] を用いてこれら 2 種類のノードを製作し、3×3 セルのベースコストをあらかじめ測定して経路探索を行った。フィールドマップにおける 1 つのセルの大きさは実際に作成したノードの大きさ相当とし、20cm 四方とする。

##### 4.1 従来手法における経路探索結果の比較

筆者らはこれまでに、ノードの移動特性および経路上の障害物を考慮した最小コスト経路探索アルゴリズムを提案し、人工知能の分野で経路探索に広く使用されている A\* アルゴリズムとの比較を行った。A\* アルゴリズムは、途中で通過する経路のコストを考慮して最短経路の探索を行うためのアルゴリズムである。A\* アルゴリズムでは、経験的に最終状態に近づく経路を優先的に探索していくことで計算量を軽減し、解となる経路がある場合には必ず最短経路が得られることが証明されている [1]。

図 5 に、A\* アルゴリズム、戦車型ノードを想定した提案アルゴリズム、および車型ノードを想定した提案アルゴリズムによる経路探索結果の一例を示す。A\* アルゴリズム (図 5(a)) は



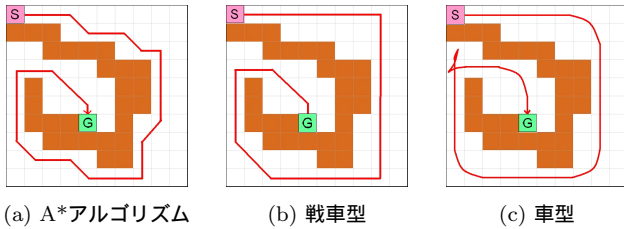


図 5 経路探索結果例

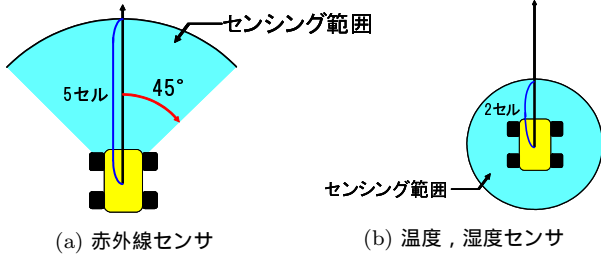


図 6 センシングモデル

表 3 センサの搭載情報

ノード	種類	搭載するセンサ
1	車	温度
2	キャタピラ	湿度
3	車	赤外線

ゴール地点までの最短経路を求めが、最短経路が複数ある場合、ゴール地点により早く近づく経路を優先的に選択するため、このように方向転換する回数が多い移動経路が得られる場合がある。しかし戦車型ノードは方向転換するごとに停車、旋回、前進を行うため、方向転換が少ない場合に比べて消費電力が増加したり、ノードの移動誤差により指定された経路から外れやすくなる。提案アルゴリズムで戦車型ノードを想定した結果(図 5(b))では A\*アルゴリズムの結果と比べて、方向転換の回数が少ない経路が探索されていることがわかる。これは、提案アルゴリズムでは経路探索前に実際のベースコストを測定しており、方向転換に消費する電力を考慮した経路探索を行うためである。車型ノードを想定した場合(図 5(c))、広い道での曲がり角では、緩やかな曲線を描きながら 90° 右方向へ曲がっているが、狭い道では、いったん曲がり角を通過して停車後、後退しながら進行方向に向き、前進するという経路が選択されている。これは、車型ノードが進行方向から見て 90° 方向へは移動できないという移動特性が考慮されていることを意味する。

#### 4.2 衝突回避アルゴリズムの評価

提案した衝突を考慮した経路探索アルゴリズムについて、ノードのスタート時間を遅らせることでノード同士の衝突を回避する単純なアルゴリズムとの比較評価を行った。後者のアルゴリズムは、まず各ノードの目的地となりうる全ての地点までの経路を探索し、その中で移動コストが最小となる配置場所を求める。次に、全てのノードがセンシング領域に配置される組合せを列挙し、全ノードの移動コストの合計が最小となる組合せを選択する。衝突が発生する場合、衝突を避けるために優先度の低いノードは衝突が起こらなくなるまでスタート時間を遅

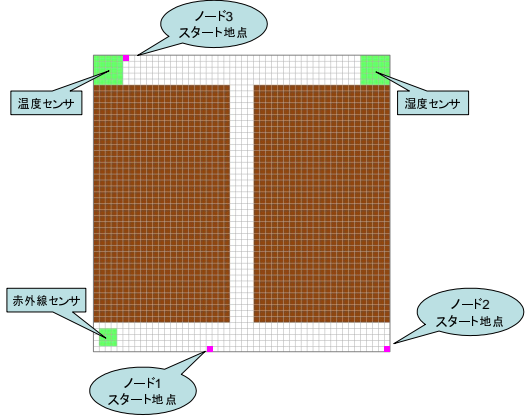


図 7 シミュレーション環境

らせる。以降、全ノードについて優先度の高いノードと衝突しないでセンシング領域に到達するまで処理を繰り返す。ノードはスタート地点での待機も含めて停止中も待機電力を消費する。待機電力は、実機により測定した値を用いる。なお、待機電力量は同じ時間移動するのに消費する電力量の約半分であった。ノードは戦車型、車型の 2 種類を使用し、温度、湿度、赤外線の 3 種類のセンサを搭載してセンシングを行う。3 種類のセンサのセンシング範囲は実際のセンサの特性を考慮して図 6 のように定義した。

#### 経路探索結果の比較

まず提案アルゴリズムの動作をわかりやすく説明するため、図 7 に示す通路型の地形コスト配置において 3 台のノードでのセンシングを想定したシミュレーションを行った。マップの広さは 10m 四方であり、地形コストは無量大としてノードが通過できないものとする。以下の問合せ構文に示すように、温度、湿度、赤外線センサによるセンシングがそれぞれ異なる場所で要求されているとする。

```
SELECT temp, 1
FROM (0, 0), (5, 5)
SELECT humid, 1
FROM (45, 0), (5, 5)
SELECT infrared, 1
FROM (1, 46), (3, 3)
```

この要求に対して戦車型ノード 1 台と車型ノード 2 台を用い、各ノードは 1 箇所のセンシングを行う。なお、ノードの番号が小さいほど優先度が高く、衝突が生じる時には優先度の高いノードの経路が優先され、優先度の低いノードが経路を変更する。使用するノードが搭載するセンサの種類を表 3 に示す。図 7 に示す通路型の地形において、ノードがマップの上部から下部へ、または下部から上部へ移動するには中心の通路を通らなければならない。センシング要求の内容と 3 台のノードが搭載するセンサの種類から、ノード 1 はマップ下部からマップ左上部へ、ノード 2 はマップ右下部からマップ右上部へ、ノード 3 はマップ左上部からマップ左下部へそれぞれ移動する。

提案手法および停止アルゴリズムにより得られたノードの経

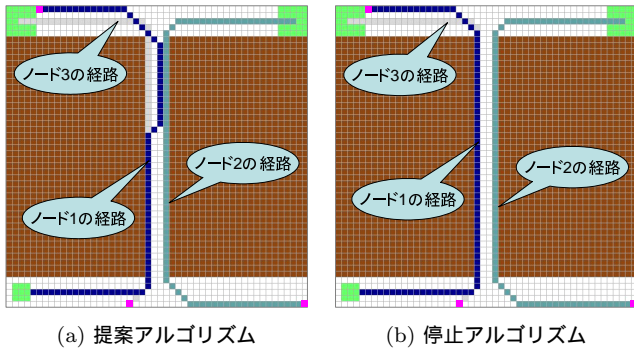


図 8 経路探索結果

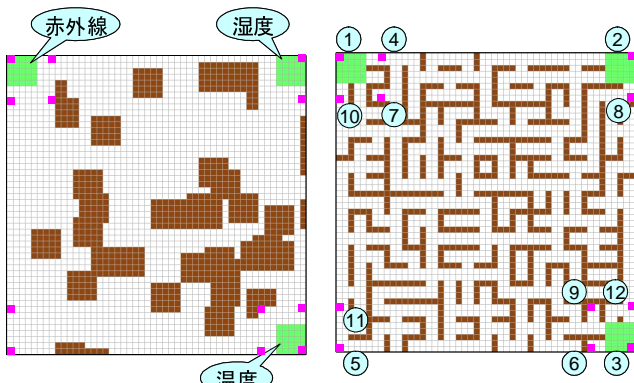


図 9 ランダム配置例

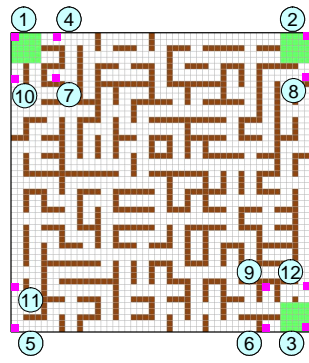


図 10 迷路状配置例

路探索結果を、図 8(a)、図 8(b) に示す。停止アルゴリズムはノードのスタート時間を遅らせることで衝突を回避するため、経路は従来手法と変わらない。従来手法を用いた場合、図 8(b) に示すようにノード 1 とノード 3 の経路が途中で交錯して衝突してしまうため、ノード 3 は衝突を回避するためにノード 1 が自身の経路上から外れるまでスタート時間を遅らせなければならない。また、他のノードが進行方向から直線的に移動してくる場合、相手のノードが自身の進もうとする経路から外れるまでスタート時間を遅らせなければならない。また、待機電力を余分に消費したり、全てのノードがセンシング領域に配置されるまでの時間が長くなる。また、従来のアルゴリズムで探索された経路上で他のノードが停止してセンシングを行う場合、ノードがセンシング領域に到達できないこともある。

提案手法では、マップ下部からはノード 1 と 2 が上部へ移動するため、ノード 3 の経路は 2 台との衝突を回避しつつ移動コストを低く抑える経路が選択されている。まず、先に下部から移動してくるノード 1 との衝突を回避し、さらにノード 1 を追って移動してくるノード 2 との衝突を回避する。提案手法では、経路探索の過程において優先度の高いノードが通過する時間と計算中のノードが通過する時間を比較し、衝突する場合はその経路の以降の探索を中止して衝突しない他の経路を探索するため、ノードが通過する道がある程度広ければ、図 8(a) に示すように、余分に電力を消費することなくノード同士がすれ違う経路を探索できる。

地形コスト配置の変化による消費電力および消費時間の比較

地形コスト配置を変化させてノード 1 台あたりの消費電力お

表 4 使用するノード

ノード	種類	搭載するセンサ	ノード	種類	搭載するセンサ
1	戦車	温度, 湿度	7	車	温度, 湿度
2	戦車	温度, 赤外線	8	車	温度, 赤外線
3	戦車	湿度, 赤外線	9	車	湿度, 赤外線
4	戦車	温度	10	車	温度
5	戦車	湿度	11	車	湿度
6	戦車	赤外線	12	車	赤外線

表 5 地形コスト配置の変化による消費電力

	ランダム	迷路
停止アルゴリズム	365.4[J]	633.1[J]
提案手法	228.5[J] (37%)	350.7[J] (44%)
理論値	208.8[J] (43%)	303.9[J] (52%)

表 6 地形コスト配置の変化による消費時間

	ランダム	迷路
停止アルゴリズム	1059.5[s]	1531.9[s]
提案手法	363.3[s] (65%)	546.7[s] (64%)
理論値	319.2[s] (69%)	518.4[s] (66%)

よび消費時間の比較を行った。ここで、消費電力はノード 1 台が移動に要する消費電力量の試行回数による平均であり、消費時間は、全ノードが同時に移動し始めてから担当するセンシング領域へ移動し終えるまでの時間の試行回数による平均を示す。20m 四方のフィールドマップ上に以下に示す 2 通りの地形コストを配置し、経路探索を行った。

- 正方形の地形コストをランダムに配置 (図 9)
- 迷路 (図 10)

ノードは、戦車型および車型をそれぞれ 6 台ずつ、計 12 台利用し、表 4 に示すように各ノードには 2 種類または 1 種類のセンサを搭載する。ノード ID が小さいほど優先度が高いものとし、優先度が低いノードは優先度が高いノードと衝突を回避するような経路が探索される。図 9 に示すように、センシング領域はフィールドマップの隅 3 箇所とし、温度、湿度、赤外線の 3 種類のセンサによるセンシングが要求されている。また、ノードの初期位置はそれぞれ図 10 に示すように設定した。このような環境下で提案手法、比較手法 (停止アルゴリズム) を用いてそれぞれ 10 回ずつシミュレーションを行った。

表 5 は 2 種類の地形コストを想定した経路探索により得られたノード 1 台あたりの消費電力を示す。ここで理論値は、同一セルに複数ノードが重なってもノード同士が衝突しないと仮定し、従来のアルゴリズムにより経路を求めた結果である。また、提案手法の結果および理論値における括弧内の値は比較手法の結果に対する改善率を示す。地形コストの配置に関わらず、提案手法は従来のアルゴリズムにより得られた結果を単純に足し合わせた理論値と比較して若干電力を余分に消費するが、比較手法よりもノードの移動に要する消費電力が低く抑えられている。比較手法では、いったんノードが動き始めると停止することなくゴール地点に到達できるようにスタート時間を遅らせるため、初期位置で待機している間にノードは待機電力を消費する。

一方、提案手法では他のノードとの衝突を回避すると同時になるべく最小コストとなる経路を探索するため、結果的に比較手法よりも消費電力を低く抑えられている。

また、地形コストを迷路状に配置した場合、ランダムに配置した場合の結果と比較して比較手法からの提案手法の改善率が大きい。迷路状に地形コストが配置された環境では、ノードが移動するための経路が複数通り存在する一方で、通路の幅がランダム配置の場合と比較して狭いため同時に通過できるノード数が限定される。そのため、最小コストとなる経路が選択される従来手法で得られる経路では衝突が起こりやすく、比較手法を用いた場合優先度がより低いノードは優先度の高いノードが自分の経路上からそれまで待機する必要がある。特に進行方向から移動してくるノードと経路が重複する場合、ノードはスタート時間を長時間遅らせる必要がある。提案手法では優先度が低いノードは他のノードと衝突しない迂回する経路を選択するため多少電力を余分に消費するが、他のノードが通過するまで待機する必要が無いため、結果的に比較手法よりも消費電力が低く抑えられる。

表6は経路探索により得られた、全ノードの配置が完了するまでの消費時間を示し、提案手法の結果および理論値における括弧内の値は比較手法の結果に対する改善率を示す。消費電力の結果と同様に、地形コストの配置に関わらず提案手法の方が比較手法よりも消費時間が短縮されている。比較手法では他のノードと衝突しなくなるまでスタート時間を遅らせるのに対し、提案手法では他のノードとの衝突を回避しながらも、方向転換の回数がより少なく、ノードがよりスムーズに通過できる経路を探索するため消費時間が短くなっている。現在のシミュレーションでは想定していないが、あるセンシング領域のセンシングに対して制限時間が指定されている場合、従来手法を組み合わせた比較手法では制限時間以内にノードが到達できずに対応できないような要求にも提案手法では対応できる可能性が高くなる。

## 5. 考察

### 5.1 ノードの巡回について

現在は、ノードはスタート地点からセンシング領域へ移動した後は継続的にセンシングし続け、他のセンシング領域へは移動しないものとしている。しかし、移動型ノードを用いたセンサネットワークでは、ノードは複数個所のセンシング領域を巡回して連続的にセンシングを行うことがある。そこで、現在は一つのノードにおいてスタート地点から1箇所目のセンシング領域への移動経路を求めているが、2箇所目、3箇所目というように複数個所のセンシング領域を循環する順番を全て列挙し、それぞれについて提案するアルゴリズムにより経路探索を行った上で、他のノードの計算結果との組合せのなかから、消費電力の合計が最小となる経路を決定することを考えている。

### 5.2 計算時間の問題

前述したようにノードが複数のセンシング箇所を巡回することも同時に想定する場合、計算時間が著しく増加する。まず、ノード毎に担当するセンシング領域の数および組合せが複数パ

ターン列挙する。次にスタート地点からあるセンシング領域、あるセンシング領域から次のセンシング領域までの経路探索を行うが、センシング領域には複数の配置場所候補が存在し、それらの組合せ数計算を行わなければならない。さらに1つの組合せにおいてノードが巡回する順番も複数通り考えられ、最後に他のノードとの分担パターンを列挙しなければならない。これらを総当りで計算すると膨大な時間を要すると考えられ、実用には適さない可能性がある。

この問題を解決するため、何らかの枝狩り処理を行うことを検討する。具体的には、あるセンシング領域からあるセンシング領域への経路探索については代表的な経路を一本計算し、その消費電力の結果を利用する。ノードが担当するセンシング領域およびそれらのセンシング領域を巡回する順番の組合せの中で全てのノードが消費する消費電力量が最も低くなるものを見当をつける。

### 5.3 考慮できていない実用上の問題

本稿で提案する方式を用いることで、これまでに実用上問題であったノード同士の衝突を回避した経路探索が可能となった。一方、下記のように未解決の問題も多く残っている。

現在ベースコストは、周辺のセルへ手動でノードを移動させることで測定している。そこで、ユーザの負担を減らすべく自動的に測定できるプログラムの作成を検討している。また、現在経路探索に使用しているベースコストにおけるセルの大きさは、ノード毎に最適な大きさを決定している。今後さらに様々なノードの利用に対応するべくベースコストのセルの大きさをさらに小さく区切り、大きさの異なるノードを統一的に扱う手法を検討する。

実機は、本稿で提案したアルゴリズムにより得られた経路に沿って移動する際に、タイヤのスリップや地形の細かい変化、さらには使用するノード自体の動作精度の低さにより、指定された経路と実際にノードが通過する経路に誤差が生じる。そのため、ノードの位置を推定し、移動誤差が生じていれば指定された経路上へノードを移動させなければならない。ノードの位置推定の手法としては、加速度センサやジャイロセンサといった内界センサを利用するデッドレコニングと、カメラなどの外界センサを用いる手法に大別できる。外界センサは一般的に高価であり、センサの分解能やノイズ等により誤差が常に生じる恐れがある。一方、内界センサはノードに複数搭載することでより高い精度の位置推定が可能となるが、ノードが大型になったり高価になる。

ノードの位置推定に関しては、複数の光学マウスセンサを利用して移動距離を調節床面から読み取ることで位置推定を行う研究[5]がなされており、車輪の変形やスリップの影響をなるべく受けにくい高精度のデッドレコニングを実現している。そこで、本研究においてもノードに安価な内界センサを搭載することを考えている。

### 5.4 問合せ言語の検討

本稿で提案した問合せ言語では、移動型ノードを利用したセンサネットワークで使用するための従来の問合せ言語からの改良点としてセンシング精度の要素を新たに設けている。今後、

ノードがセンシング領域へ移動後さらに異なるセンシング領域へ移動することを想定すると同時に、問合せ言語をさらに改良することを考えている。具体的には、センサデータの閾値によるさらに詳細な条件設定が考えられる。別の問合せ結果を元に、次に要求するセンサデータを決定する。

また、現在は、1台のノードが1箇所のセンシングを行うことを想定しているが、センシング精度を上げるための手段としては複数のノードを同時に配置させることも考えられる。そこで、現在のセンシング精度をさらに細かく設定できるようにし、センシングを行う台数等の条件を指定できるようにすることを考えている。

#### 5.5 通信範囲の問題

移動型ノードを用いたセンシングにおいて、現在はノードがセンシング領域へ移動する段階に着目して研究を進めており、センシングを行った後にデータセンタにセンサデータを転送することを考えていない。実際には、センサデータに制限時間が設定されるとデータセンタまでデータを制限時間内に到達させなければならないため、ルーティングプロトコルについても検討しなければならない。しかしこの問題に対しては、センシング領域へ移動した後データセンタまで戻ってくる経路を探索することで解決できると考えている。もし、計算結果から、ノードがセンシング後にデータセンタへ帰ってくると指定された時間に間に合わない場合でも、センサデータを転送することを念頭において、あらかじめ必要なノードを転送可能な位置に配置させることで対処できる。

### 6. 関連研究

これまでの移動型センサネットワークに関する研究としては、複数のノードが協調して様々な状況に応じたルーティングを行うことを目的とした RAMOS [7] や、ノードの正確な位置特定を目的とした研究 [8]、さらにセンシングカパレッジの拡大を目的とした研究 [9] が行われている。しかしこれらの研究では、本研究で提起した実用上の問題が考慮されていない。

また、ノードが人や動物に寄生して移動するという新しい移動手法を提案し実際に実機の製作を行っている例として、Parasitic Mobility [3] が挙げられる。しかし Parasitic Mobility は寄生対象が常に存在する環境を想定しており、人や動物が近づけない危険地帯へのノードの配置は不可能であるため、移動型センサネットワークの特徴を活かしきれていない。

今枝らは、センサノードに関する専門的な知識がないユーザでもアプリケーションを容易に利用できるようにする、ROASEN システムを提案している [2]。アプリケーション開発者はセンサノードの配置条件を XML 形式で記述することで、センシング目的に応じたノード配置が可能となる。しかし、XML ではノードの配置場所について、オブジェクトや壁からの距離など詳細な条件を設定する必要があり、ユーザの負担となる。また、アプリケーション毎に XML を記述しなければならず、あまり汎用的ではない。本稿で提案する手法では、問合せ言語を用いてセンシング領域やセンサの種類等を指定するだけで、ノードの配置場所等も自動的に決定できる。

### 7. 結論

本稿では、筆者らがこれまでに提案してきたコストマップを用いた経路探索アルゴリズムでは考慮できなかったノード同士の衝突の問題を解決する、複数ノードの衝突を考慮した経路探索アルゴリズムを提案した。これにより、衝突を効率的に回避する経路探索が可能となった。さらに、移動型ノードを用いたセンサネットワークのための問合せ言語を提案した。ユーザは、センシングしたい場所、センサの種類などの条件を容易に設定でき、複数ノードをセンシングの目的に応じて効率的かつ自動的に配置できるようになった。本研究では、スタート時間を遅らせることで衝突を回避するアルゴリズムと提案手法をシミュレーションにより比較し、提案手法の有効性を示した。

今後は、ノードがセンシング領域へ移動後さらに別のセンシング領域へ移動することを想定した最小コスト経路探索アルゴリズムを考案する予定である。

謝辞

本研究の一部は、文部科学省 21 世紀 COE プログラム「ネットワーク共生環境を築く情報技術の創出」、および文部科学省基盤研究 (A) (17200006) の研究助成によるものである。ここに記して謝意を表す

#### 文献

- [1] D. M. Bourg and G. Seemann, "AI for Game Developers," O'Reilly, Jul. 2004.
- [2] 今枝卓也, 大澤 亮, 高汐一紀, 徳田秀幸, "アプリケーション適応型センサノード配置ロボットの提案", 情報処理学会研究報告 (コピキタスコンピューティングシステム研究会 2006-UBI-11), Vol. 2006, No. 119, pp. 29-35, Nov. 2006.
- [3] M. Laibowitz and J. A. Paradiso, "Parasitic mobility for pervasive sensor networks," Proc. of Third International Conf. on Pervasive Computing (PERVASIVE 2005), pp. 255-278, Munich, May 2005.
- [4] 中宮正樹, 岸野泰恵, 寺田 努, 西尾章治郎, "コストマップを用いた移動型センサノードの経路探索手法," 電子情報通信学会研究報告 (センサネットワーク時限研究専門委員会), pp. 107-114, Tokyo, May 2006.
- [5] 関森大輔, 宮崎文夫, "複数の光学マウスセンサを用いた移動ロボットのデッドレコニング," 計測自動制御学会論文集, Vol. 41, No. 10, pp. 775-782, Oct. 2005.
- [6] G. T. Sibley, M. H. Rahimi, and G. S. Sukhatme, "Robomote: A tiny mobile robot platform for large-scale ad-hoc sensor networks," Proc. of the IEEE International Conf. on Robotics and Automation (ICRA 2002), pp. 1143-1148, Washington, May 2002.
- [7] R. Suzuki, K. Makimura, H. Saito, and Y. Tobe, "Prototype of a sensor network with moving nodes," Proc. of 1st International Workshop on Networked Sensing Systems, Tokyo, Jun. 2004.
- [8] S. Tilak, V. Kolar, N. B. Abu-Ghazaleh, and K. D. Kang, "Localization for mobile sensor networks," Proc. of International Conf. on Mobile Computing and Networking, pp. 45-57, Philadelphia, Sep. 2004.
- [9] G. Wang, G. Cao, and T. L. Porta, "Movement-assisted sensor deployment," Proc. of the Conf. on Computer and Communications Societies (INFOCOM 2004), Vol. 23, No. 1, pp. 2469-2479, Hong Kong, Mar. 2004.
- [10] Crossbow, <http://www.xbow.jp/>.
- [11] MindStorms, <http://mindstorms.lego.com/>.