

領域被覆のためのセンサネットワークアルゴリズム

森川雅和, 鈴木朋子, 大下福仁, 角川裕次, 増澤利光

大阪大学大学院情報科学研究科コンピュータサイエンス専攻

センサネットワークの利用例の多くに, ある特定の領域の監視を行い, 得られた情報を収集したいという要求がある. 各センサの検知領域で監視を行いたい領域(要求領域)を被覆する連結なセンサの集合を連結センサカバーとよび, センサ数が最小の連結センサカバーを求めることは NP 困難であると知られている. 分散環境において, 少数のセンサで構成される連結センサカバーを求める手法が既に提案されているが, 既存手法ではアルゴリズムを開始するのはただ 1 つのセンサであり, 広大な要求領域に対し連結センサカバーを求めるには非常に長い時間を要する. そこで本研究では, 分散環境において要求領域の被覆を複数センサで平行に行う手法を提案する. また実験により, 提案手法を用いて短時間で十分に少ないセンサ数の連結センサカバーが得られることを示す.

A Connected Sensor Cover Algorithm in Sensor Networks

Masakazu MORIKAWA, Tomoko SUZUKI, Fukuhito OOSHITA,
Hirotsugu KAKUGAWA, Toshimitsu MASUZAWA

Graduate School of Information Science and Technology, Osaka University

A *connected sensor cover* is extremely useful to monitor an area in sensor networks: a connected sensor cover is a sub set of connected sensors, which covers the area to be monitored with the sensors' sensing regions. It is known that the *connected sensor cover problem*, which requires to find a connected sensor cover with the smallest number of sensors, is NP-hard. A connected sensor cover algorithm was proposed. However, the larger the area to be monitored is, the more it takes to find a connected sensor cover since only one sensor starts the algorithm. Therefore, in this paper, we proposed an algorithm, in which multiple sensors concurrently elect sensors to be included in a connected sensor cover. We show by simulation that the proposed algorithm finds a connected sensor cover with small number of sensors within a sufficiently short time.

1 はじめに

センサネットワークとは, 小型自律センサで構成される無線通信ネットワークである. 各センサは検知領域内に存在する対象物の観測を行うことができ, 通信領域内に存在するセンサと無線通信によるメッセージ交換が可能である. そのため, センサネットワークは戦場における特定領域に存在する戦車の位置の捕捉や, 農園における特定領域の温度管理などへの適用が期待されている. このようにセンサネットワークを用いてある特定の領域を監視したい場合, 通常, 監視を行いたい領域に多数のセンサが配布され, 各センサが自身の検知領域内の監視を行い, 各センサが検知して得た情報を収集し処理を行う. このとき, ある特定の監視を行いたい領域(要求領域)を各センサの検知領域で被覆し, さらに要求領域の検知を行っている任意のセンサは通信可能(連結)である必要がある. このように, 要求領域をセンサの検知領域で被覆する連結なセンサの集合を連結センサカバーとよぶ. 領域に配布した多くのセンサを用いれば連結センサカバーを構成できるが, 電力消費などの観点から, 連結センサカバーは出来るだけ少数のセンサで構成されることが望ましい. しかし, センサ数最小の連結センサカバーを求めることは非常に困難(NP 困難)であることが知られている.

連結センサカバーを求める手法は既に多く研究されている [3, 4, 5]. Himanshu ら [2] は, $O(\log n)$ -近似アルゴリズム (n はセンサ数) を提案し, さらにこの逐次アルゴリズムを拡張し, 分散環境へ適用している. Himanshu らの手法を用いることで, 分散環境においても少数のセンサで連結センサカバーを求めることが可能である. しかし Himanshu らの手法では, ネットワーク中のただ 1 つのセンサがアルゴリズムを開始し, 選出されているセンサの周辺の中から要求領域を効果的に被覆するセンサを選出することを繰り返し, 徐々に連結センサカバーに含めるセンサを求めていく. そのため, センサの検知領域に対し要求領域が広大な場合, 連結センサカバーを求めるまでに要する時間が増大してしまう.

そこで本研究では, 複数のセンサがアルゴリズムを開始し, 複数のセンサが平行して連結センサカバーを求める手法を提案する. これにより, 連結センサカバーを求めるまでに要する時間の短縮を図った. 提案手法では個々のセンサが連結センサカバーに含めるセンサを独立に選出する. そのため各センサが Himanshu らの手法を用いて選出を行うと, ある同じ領域を被覆する複数のセンサが異なるセンサによって選出され, 過大なセンサ数の連結センサカバーが構築されてしまう. そこで本研究では, Himanshu ら

の手法を改良し、求める連結センサカバーに含まれるセンサ数の増加を抑えながら複数のセンサが効果的に連結センサカバーに含めるセンサを選出する手法を提案する。本論文では、シミュレーションにより、提案手法が Himanshu らの手法に比べ連結センサカバーを求めるまでに要する時間が短いことを示す。またこのとき、連結センサカバーに含まれるセンサ数の増加が抑えられており、さらに被覆に用いたメッセージ数を削減していることも示す。

本論文では、2章でシステムモデルの説明と問題定義を行う。3章、4章ではそれぞれ既存手法と提案手法の説明を行い、5章にてシミュレーションによる評価を行う。最後に6章でまとめと今後の課題について言及する。

2 システムモデルと問題定義

2.1 システムモデル

センサネットワーク $G = (V, E)$ は、センサ集合 $V = \{I_1, I_2, \dots, I_n\}$ (n はセンサ数) とリンク集合 E で定義される。各センサ I_i にはそれぞれ固有の識別番号 ID_i が割り振られている。また、各センサ I_i は二次元平面 R 上に配置されており、自身の地理的位置を知っている。各センサ I_i は自身を内部に含む通信領域 T_i を持ち、その中に存在する他のセンサとのみ無線通信が可能である。センサ I_i, I_j が互いに通信可能であるとき、 G 上の I_i, I_j 間に双方向通信リンク $e_{ij} \in E$ が存在し、 I_i, I_j は隣接するという。本研究では隣接するセンサ I_i, I_j 間のみでメッセージの送受信が可能であるとする。すなわち、センサ I_i はセンサ I_j と通信可能であるが、センサ I_j はセンサ I_i と通信不可能な場合、 I_i, I_j 間ではメッセージの送受信を行わない。また、ネットワーク中のセンサはすべて起動しており、センサ I_i がセンサ I_j に送信したメッセージはセンサ I_j が必ず受信する。このとき、通信の衝突などによるメッセージの消失、およびメッセージの改変は起こらないとする。センサ I_i, I_j 間はメッセージ交換によってのみ情報交換を行う。センサ I_i の検知領域 S_i とはセンサ I_i が平面 R 上の監視を行える領域であり、検知領域 S_i はセンサ I_i を内部に含む。また、センサネットワーク G において、互いの検知領域に共通領域が存在する任意のセンサ I_i, I_j の最大通信距離をリンク半径 r_G とよぶ。本論文では、領域 X の面積を $|X|$ と表す。

2.2 連結センサカバー問題

センサネットワークでは、監視対象の領域をセンサの検知領域で被覆したいという要求が多々ある。このとき、センサは限られたバッテリーしか搭載していないため、より少ないセンサで監視対象の領域を被覆することが重要である。また、一般的に監視対象の領域から収集したデータを伝搬するために、各センサが互いに通信可能であることも求められる。そ

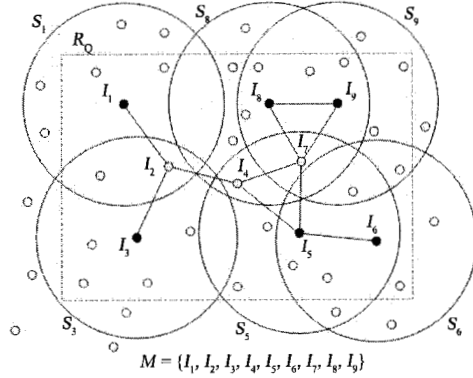


図 1: 連結センサカバーの例

ここで本研究では、連結センサカバー問題を考える。

連結センサカバー問題とは、センサネットワーク G 、および被覆すべき領域が与えられたとき、可能な限り少数の連結なセンサを用いて被覆すべき領域をセンサの検知領域で被覆する問題である。被覆すべき領域を要求領域 R_Q とよび、要求領域 R_Q はセンサネットワーク G が存在する二次元平面 R 上の連続する平面領域である。つまり、要求領域 R_Q が共通領域を持たない 2 つ以上の平面領域で与えられることはない。連結センサカバーを以下に定義する。

定義 2.2.1 (連結センサカバー問題)

センサネットワーク $G = (V, E)$ に存在する各センサ I_i の検知領域を S_i とする。センサネットワーク G に対して要求領域 R_Q が与えられたとき、以下の条件を満たすセンサの集合 $M = \{I_{i_0}, I_{i_1}, \dots, I_{i_m}\} (\subseteq V)$ を連結センサカバーとよぶ。

1. $R_Q \subseteq (S_{i_1} \cup S_{i_2} \dots \cup S_{i_m})$
2. G の M による部分グラフは連結である

連結センサカバー問題とは、サイズが最小の連結センサカバー M を求める問題である。

ここで、線分を用いた点被覆問題より一般的でない問題は NP 困難であることが知られており、連結センサカバー問題もまた NP 困難として知られている [1]。

本論文では、センサ数 n は連結センサカバーを構築するのに十分に大きいとする。すなわち、要求領域 R_Q を被覆可能な連結センサカバー M は必ず存在する。また、各センサ I_i には要求領域 R_Q の情報が与えられているとする。

図 1 に連結センサネットワークカバーの例を示す。センサ集合 $\{I_1, I_2, \dots, I_9\}$ は連結センサカバーである。センサ $I_1, I_3, I_5, I_6, I_8, I_9$ の検知領域で要求領域 R_Q を被覆し、センサ I_2, I_4, I_7 によって集合に含まれるセンサ間の連結性を保っている。

3 Himanshu らのアルゴリズム

本研究では、複数センサが連結センサカバーアルゴリズムを開始し、要求領域 R_Q を被覆する手法を提案する。各センサが実行する連結センサカバーアルゴリズムには、Himanshu らの手法を改良したものを用いている。Himanshu らは連結センサカバーを求める逐次アルゴリズムを提案し、それが $O(\log n)$ -近似アルゴリズムであることを示した [2]。また、その逐次アルゴリズムを分散環境であるセンサネットワークに適用し、連結センサカバーに含まれるセンサ数が逐次アルゴリズムの場合とほぼ変わらないことをシミュレーションにより示した。そこで本章では、分散環境における Himanshu らのアルゴリズムの概要を説明する。

Himanshu らのアルゴリズムでは、要求領域内の任意の 1 センサのみがアルゴリズムを開始し、周辺のセンサを徐々に連結センサカバーを構成するセンサとして選出していく。連結センサカバーを構成するセンサとしてまだ選出されていない周辺センサを新たにカバーに加える動作をした回数を k とすると、すでに選出されたセンサ集合をカバー M^k と表す。アルゴリズム開始直後においては、アルゴリズムを開始した 1 センサのみがカバー M^0 に含まれる。また、カバー M^k に含まれるセンサの検知領域で被覆している要求領域 R_Q をカバー M^k の検知領域とよび、 $S(M^k)$ と表す ($S(M^k) = \bigcup_{I_i \in M^k} (S_i \cap R_Q)$)。カバー M^k において次にどの周辺センサを連結センサカバーに加えるかは、カバー M^k に含まれる 1 センサが決定する。このセンサをカバー M^k のコーディネータ C^k とよぶ。アルゴリズム開始直後では、 M^0 に含まれるただ 1 つのセンサがコーディネータ C^0 となる。コーディネータとなるセンサは、カバーにセンサが追加される毎に変わり、 $C^{k_1} \neq C^{k_2} (k_1 \neq k_2)$ である。

カバー M^k において、コーディネータ C^k は検知領域 $S(M^k)$ 周辺のセンサから複数センサを選出し、カバー M^{k+1} に追加して連結カバーを構成するセンサを増やしていく。以下、その方法を述べる。コーディネータ C^k は、周辺センサに探索メッセージを送信し、周辺にどのようなセンサが存在するのかを確かめる。得られた情報に基づき、コーディネータ C^k は効果的に要求領域を被覆できるセンサを複数選出する。このとき、センサの検知領域と要求領域の共通領域を個々に評価してカバーに加えるセンサを選出すると、多くのセンサの検知領域が同一領域を被覆してしまう可能性がある。そこで、探索メッセージを受信したセンサ I_j に対し、 I_j からコーディネータ C^k までの通信経路の 1 つをセンサ I_j の候補パス P_j^k とする。コーディネータ C^k は被覆されていない要求領域の広い範囲を少数のセンサで被覆する候補パスを以前のコーディネータが得た候補パス

も含めた中から選出する。そして、選出された候補パス上に存在するすべてのセンサをカバー M^{k+1} に加える。

カバー M^k における Himanshu らのアルゴリズムを“探索フェーズ”、“応答フェーズ”、“決定フェーズ”の 3 つのフェーズにわけて説明する。

探索フェーズ 探索フェーズでは、コーディネータ C^k が周辺センサへ探索メッセージを送信する。探索メッセージにはカバー M^k の検知領域 $S(M^k)$ 、およびコーディネータ C^k が送信者である情報を付加する。探索メッセージはフラッディングによってコーディネータ C^k から $2r_G$ ホップ以内に存在するすべてのセンサに転送される。すなわち、コーディネータ C^k の検知領域と自身の検知領域が共通領域をもつセンサ $I_i (S_i \cap S_j \neq \phi (C^k = I_h))$ と、さらにその S_i と検知領域に共通領域をもつセンサ $I_j (S_i \cap S_j \neq \phi)$ すべてに探索メッセージが伝達される。各センサ I_i は、コーディネータ C^k からの探索メッセージは初めに受信したもののみを受信し、その探索メッセージが経由した経路がセンサ I_i の候補パス P_i^k となる。

応答フェーズ 探索メッセージを受信したセンサ I_i がコーディネータ C^k へ応答メッセージを返す。ただし、応答メッセージを返すのは、自身の検知領域 S_i とカバー M^k の検知領域 $S(M^k)$ が共通領域を持つ場合のみである ($S_i \cap S(M^k) \neq \phi$)。この応答メッセージは、探索メッセージが経由した経路、すなわち候補パスを逆に辿ってコーディネータ C^k まで伝えられる。応答メッセージには候補パス (パス上に存在するセンサ識別子の列) と、パス上の各センサの検知領域の情報を付加する。

決定フェーズ 決定フェーズでは、応答フェーズで得られた各候補パスの情報からコーディネータ C^k はカバー M^k に追加すべき候補パスを選出し、選出された候補パス上のすべてのセンサをカバー M^k に追加する。コーディネータ C^k は、応答メッセージによって得られた各候補パス P_i^k を候補パス集合 SP^k に追加し、さらに各候補パス P_i^k の検知領域 $S(P_i^k)$ を保持しておく。候補パス P_i^k の検知領域 $S(P_i^k)$ は以下で求められる。

$$S(P_i^k) = \bigcup_{I_j \in P_i^k} (S_j \cap R_Q)$$

連結センサカバー問題では、少数のセンサのできる限り広い範囲の要求領域を被覆したい。そこでコーディネータ C^k は、すべての応答メッセージを受信すると、候補パス集合 SP^k に含まれる候補パスの中から以下で表される候補パスの評価値が最大となるパスを選出する。

$$\text{候補パス } P_i^k \text{ の評価値} = \frac{|S(P_i^k) \cap \overline{S(M^k)}|}{P_i^k \text{ 上に存在し、かつ} \\ \text{カバー } M^k \text{ に含まれていないセンサ数}}$$

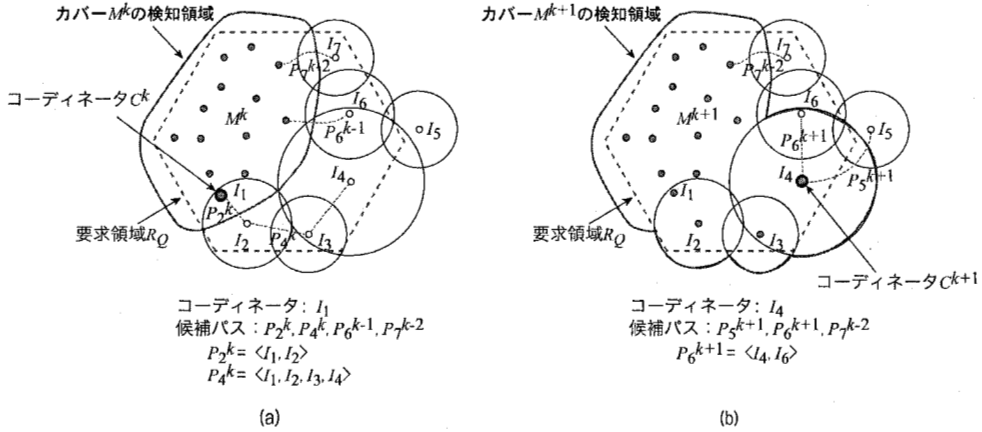


図 2: 既存手法の動作例

評価値が最大となったパスを P_i^k とする。コーディネータ C^k は決定メッセージを P_i^k 上のセンサに送信し、 P_i^k 上のセンサは新たに連結センサカバーを構成するセンサになる。すなわち、カバー M^{k+1} はカバー M^k に含まれるセンサと、新たに追加した P_i^k 上のセンサで構成される。また、センサ I_i がカバー M^{k+1} のコーディネータ C^{k+1} となる。決定メッセージには候補パス集合 SP^k 、および SP^k に含まれる各候補パスの検知領域の情報を付加し、コーディネータ C^{k+1} は候補パス集合 SP^k を自身の候補パス集合 SP^{k+1} の初期値とする。

以上のフェーズをカバー M^{k+1} の検知領域が要求領域 R_Q を被覆するまで繰り返す。

アルゴリズムの動作例を図 2 を示す。図 2.(a) では、候補パス $P_2^k, P_4^k, P_6^{k-1}, P_7^{k-2}$ が存在する。候補パス P_6^{k-1}, P_7^{k-2} はカバー M^k 以前からの候補パスである。候補パスの評価値が最大となる候補パス P_4^k が選出され、そのパス上にあるセンサ全てを M^k に追加し、カバー M^{k+1} が構成される (図 2.(b))。カバー M^{k+1} のコーディネータ C^{k+1} は I_4 となる。

4 提案手法

Himanshu らの手法では、アルゴリズムを開始するセンサはただ 1 つであり、要求領域が広大な場合、要求領域の被覆に非常に長い時間を要してしまう。そこで本研究では、分散環境において複数センサでアルゴリズムを開始させ、短時間で要求領域の被覆を行う手法を提案する。

提案手法では、アルゴリズムを開始した各センサは、周辺のセンサを徐々に連結センサカバーを構成するセンサとして選出していく。センサ I_i がアルゴリズムを開始し、 I_i から始まった処理によって連結センサカバーを構成するセンサとして選出されたセンサの集合をカバー M_i^k と表す ($I_i \in M_i^k$)。ここで、

連結センサカバーを構成するセンサとしてまだ選出されていないセンサを新たにカバーに加える動作をした回数を k とする。また、カバー M_i^k のコーディネータを C_i^k と表す ($C_i^0 = I_i$)。

提案手法では、各カバー M_i^k のコーディネータ C_i^k は周辺センサから次に自身のカバーに加えるべきセンサを他のカバーとは独立して選出する。このとき、あるセンサが同時に 2 つ以上のカバーに含まれることがないように ($M_i^k \cap M_j^k = \phi (i \neq j)$)、次に加えるべきセンサを選出する。提案手法では複数カバーが同時に同一の要求領域の被覆を行うが、Himanshu らのアルゴリズムを各カバーが独立に実行すると、多くの不要なセンサが連結センサカバーに選出されてしまう。たとえば、ある 2 つのカバーがある同一の領域付近を被覆しようとしている場合、各カバーがその領域を被覆するセンサをそれぞれ選出し、結果、互いの検知領域が大きく共通領域を持つ余剰なセンサが連結センサカバーに含まれてしまう。そこで提案手法では、他のカバーとできる限り共通領域を持たないように協調して各カバーは次に加えるセンサを選出する。

また、Himanshu らの手法では、コーディネータ C^k は各候補パスの検知領域とカバー M^k の検知領域から各候補パスの評価を行うことができた。しかし複数のカバーで並列に要求領域の被覆を行う場合、不要なセンサの選出を避けるため、他のカバーが被覆した領域も考慮しなければならない。たとえば、あるセンサ I_a がカバー M_i^k の探索メッセージに対して応答した直後に、検知領域 S_a が他のカバー M_j^k の検知領域 $S(M_j^k)$ で被覆されたとする。カバー M_i^k のコーディネータ C_i^k がセンサ I_a の検知領域とカバー M_i^k の検知領域 $S(M_i^k)$ のみに基づいて次に加えるセンサを選出すると、 I_a の検知領域 S_a が既に他のカバーに被覆されているにも関わらず、 I_a がカ

パー M_i^k に加えられる可能性がある。そこで提案手法では、各コーディネータが他のカバーの検知領域も可能な限り把握するために、開始センサを含め、選出されたセンサは自分自身が選出されたことを把握し、各センサは自身が送信するすべてのメッセージに、現在自分が把握している選出されたセンサとその検知領域の情報を付加する。これにより、各カバーの検知領域の情報は徐々にネットワーク内で広がり、各センサは他のカバーの状況もある程度、把握可能となる。そして、各センサの検知領域内でのカバーにも被覆されていない領域をそのセンサの評価値とする。さらに、決定メッセージを受信したセンサは、メッセージに含まれている自身の評価値と自分で計算し直した評価値が異なる場合、カバーに自身が加わることを放棄してコーディネータが把握できなかった状況を回避する。その際に、受信した決定メッセージにある候補センサの中から、次に評価値の高いセンサへ決定メッセージを転送する。また、Himanshu らのようにコーディネータから $2r_G$ ホップも離れたセンサの評価をしたとしても、他のカバーによる要求領域の被覆により、コーディネータが求めた評価値が実際の評価値と異なる場合が多く生じてしまう。そこで提案手法では、コーディネータは探索メッセージを隣接センサのみに送信する。これによって Himanshu らの手法に比べ探索メッセージ数が減少し、さらにコーディネータが全ての応答メッセージを待つ時間を短縮できる。

以上の処理を行うことで、各コーディネータは他のカバーが被覆した領域を考慮し次にカバーに加えるセンサを選出できる。しかし、各カバーが他のカバーの検知領域を避けつつ線状に自身のカバーを拡大してしまうと、非常に小さな未被覆の領域が各カバーの検知領域間に生じてしまい、それらの領域を被覆するために多くのセンサが選出されてしまう。そこで提案手法では、各カバーはカバーの中心から円状にカバーを拡大していく。カバー M_i^k のコーディネータ C_i^k は、検知領域 $S(M_i^k)$ の地理的中心点から要求領域内で最も近いどのカバーにも被覆されていない点を求める。この点を直近未被覆点 p_i^k とよぶと、コーディネータ C_i^k は、候補センサに含まれるセンサのうち、直近未被覆点 p_i^k を検知領域に含むセンサの中から評価値が最も高いセンサを次にカバーに加えるセンサとして選出する。

要求領域 R_Q をすべて複数カバーで被覆したとしても、各センサの通信領域が検知領域に比べて十分大きくない場合、選出されたセンサの集合に含まれる任意のセンサが連結でない場合がある。そこで、各カバー M_i^k の周辺に未被覆の要求領域がなくなった際に、コーディネータ C_i^k は連結フェーズを実行する。連結フェーズでは、コーディネータ C_i^k はカバー M_i^k 及びカバー M_j^l と共通領域を持つカバー

M_j^l の中から地理的距離が最短となる 1 組のセンサ $\langle I_a, I_b \rangle$ ($I_a \in M_i^k, I_b \in M_j^l$) を選出し、センサ I_a, I_b 間に存在するいくつかのセンサをカバーに加えてカバー間を連結にする。

5 シミュレーション結果

本研究では、シミュレーションにより提案手法の評価を行った。要求領域を 100×100 の正方形に設定し、500 個のセンサをその領域内にランダムに配置する。センサの検知領域及び通信可能領域は単一円とし、検知領域の半径は 10、通信領域の半径は 10、15、20 とした。各センサはラウンド方式で動作し、各ラウンドで各センサはメッセージを受信し、メッセージに対する適切な計算を行い、計算結果としてメッセージの送信を行う。所要時間はネットワーク内のあるセンサがアルゴリズムを開始してから、連結センサカバーが完成するまでのラウンド数とする。提案手法ではアルゴリズムを開始するセンサの数が 1, 2, 4, 6, 8, 10, 12, 14, 16, 18, 20 の場合を計測し、それぞれのセンサは同時にアルゴリズムを開始する。求めた連結センサカバーのサイズ、所要時間、メッセージ数でそれぞれについて、Himanshu らの手法に対する提案手法の比率を評価した。

図 3 は、求めた連結センサカバーのサイズを評価した結果である。縦軸は求めた連結センサカバーのサイズの Himanshu らの手法に対する比率、横軸は開始センサ数である。提案手法は開始センサ数が増えるに従って求めたセンサカバーのサイズが大きくなっている。これはコーディネータの数が多くなると、同一のラウンド時に同一領域を被覆する別々のセンサが選出される可能性が上がるためである。また、通信半径が小さくなるにつれても選出されるセンサ数は増加している。通信半径が小さくなると、各カバーの検知領域の情報が各センサに伝わる速度は下がる。結果、各センサが他のカバーの検知領域を把握しきれずに、自己の評価値を実際的评价値よりも大きく計算してしまう場合が増えるからである。さ

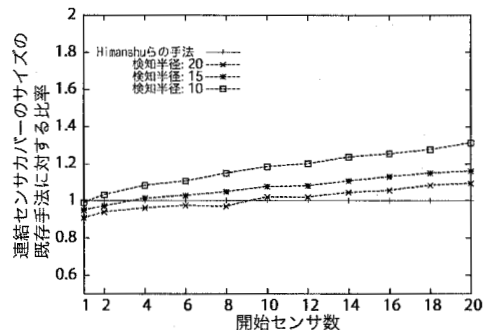


図 3: 連結センサカバーのサイズの評価結果

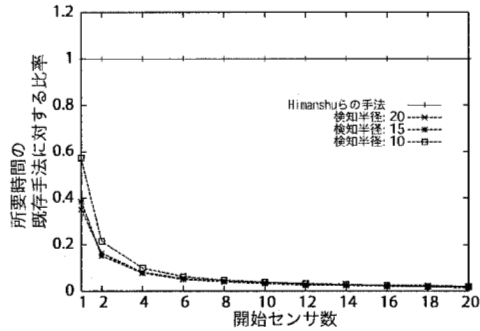


図 4: 連結センサカバー構築所要時間の評価結果

らに、各カバー間を連結するためのセンサも多く選出されるためである。ただし、いずれの場合も増加率は緩やかなものであり、開始センサが1つの場合は Himanshu らの手法よりも連結センサカバーのサイズは小さい。

図 4 は連結センサカバーを構築するまでの所用時間を評価した結果である。縦軸は連結センサカバーを構築するまでの所用時間の Himanshu らの手法に対する比率、横軸は開始センサ数である。提案手法はいずれの場合も Himanshu らの手法よりも短時間で連結センサカバーを求めていることが分かる。ここで、開始センサ数が 1 の場合でも提案手法の所要時間は Himanshu らの手法の 6 割以下である。これは各コーディネータは隣接センサからの応答を待つだけで良いので、応答メッセージを待つ時間が短縮されているからである。

図 5 は連結センサカバーを構築するまでに送信されたメッセージ数を評価した結果である。縦軸は連結センサカバーを構築するまでに送信されたメッセージ数の Himanshu らの手法に対する比率、横軸は開始センサ数である。提案手法では連結するためのメッセージが必要な場合もあるが、それらを含めてもメッセージ数は Himanshu らの手法の 0.1%~4% に収まっている。これは Himanshu らの手法では、1 つのコーディネータが送信する探索メッセージは指数的に増加していくのに対し、提案手法では常に隣接センサ数のみだからである。

以上より、提案手法は Himanshu らの手法に比べて、選出されたセンサ数の増加を抑えつつ、アルゴリズムの高速化およびメッセージコストの削減に成功している。

6 まとめ

本論文では、既存の連結センサカバーアルゴリズムを拡張し、複数センサから開始可能な連結センサカバーアルゴリズムを提案した。これは要求領域を被覆するのにかかる時間を短縮する目的からであ

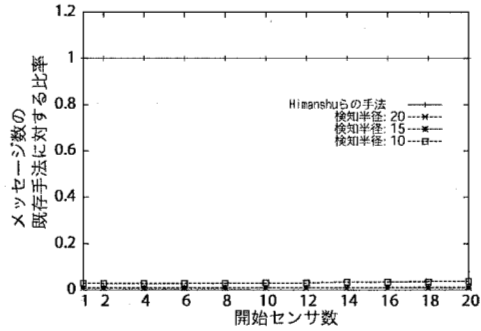


図 5: メッセージ数の評価結果

た。シミュレーションの結果より、提案手法は選出されるセンサ数の増加を抑えつつも、要求領域の被覆を高速化し、メッセージコストも大幅に削減することができた。

今後の課題としては、各センサの通信半径、検知領域が異なるシステムでの評価、センサの故障のあるシステムへの対応が考えられる。

謝辞

本研究の一部は、文部科学省 21 世紀 COE プログラム (研究拠点形成費補助金) の研究助成、日本学術振興会科学研究費補助金 (基盤研究 (B)15300017, 基盤研究 (B)17300020), 日本学術振興会特別研究員奨励費 (2005, 50673), 文部科学省科学研究費補助金 (特定領域研究 16092215, 若手研究 (B)18700059), 総務省戦略的情報通信研究開発推進制度 (SCOPE), および、大川情報通信基金によるものである。

参考文献

- [1] V. S. Anil Kumar, Sunil Arya, and H. Ramesh, "Hardness of set cover with intersection 1", *Proc. of the 27th International Colloquium on Automata, Languages and Programming - ICALP'00*, pp. 624-635, July 2000.
- [2] Himanshu Gupta, Samir R. Das, and Quinyi Gu, "Connected sensor cover: self-organization of sensor networks for efficient query execution", in *Proc. of the 4th ACM International Symposium on Mobile Ad Hoc Networking & Computing - MobiHoc'03*, pp. 189-200, Jun 2003.
- [3] Zongheng Zhou, Samir Das, and Himanshu Gupta, "Variable radii connected sensor cover in sensor networks", in *Proc. of the 1st IEEE International Conference on Sensor and Ad Hoc Communications and Networks - SECON'04*, pp. 387-396, Oct 2004.
- [4] Stefac Funke, Alex Kesselman, Zvi Lotker, and Michael Segal, "Improved approximation algorithms for connected sensor cover", in *Proceedings of 3rd International Conference on Ad Hoc Networks & Wireless ADHOC-NOW'04*, pp. 56-69, July 2004.
- [5] H Zhang, JC Ho, "Maintaining Sensing Coverage and Connectivity in Large Sensor Networks", in *Technical Report UIUCDCS-R-2003-2351, Department of Computer Science, University of Illinois at Urbana-Champaign, June 2003*