

アドホックネットワークにおける 2フェーズMaxRS検索手法 Two-Phase MaxRS Query Processing Method in Mobile Ad Hoc Networks

中山 侑紀[♡] 天方 大地[◇] 原 隆浩[◇] 西尾 章
治郎[♣]

Yuki NAKAYAMA Daichi AMAGATA
Takahiro HARA Shojiro NISHIO

本稿では、アドホックネットワークにおけるMaxRS (Maximizing Range Sum) 検索について考える。アドホックネットワークを構成する各端末が、スコアをもつ位置データをそれぞれ所持していると想定する。MaxRS 検索は、クエリ発行端末がある大きさの長方形を指定し、その長方形に包含される位置データのスコアの和が最大となる長方形の位置を取得するものである。例えば、災害地での救護活動において、救急隊員がMaxRS 検索を行い、負傷者が密集している場所を特定するといった応用例が考えられる。正確な解を検索する単純な手法は、クエリ発行端末がネットワーク内の全てのデータを収集するものである。しかし、アドホックネットワークは通信帯域に制限があり、全てのデータを収集する方法は効率的でない。そこで本稿では、アドホックネットワークにおける効率的なMaxRS 検索手法を提案する。提案手法は、1 フェーズ目に、MaxRS 検索の結果を取得するために必要なデータが存在する領域を特定する。さらに、2 フェーズ目に、特定した領域に存在するデータのみを収集することにより、不必要なデータの送信を抑制する。シミュレーション実験の結果から、提案手法はMaxRS 検索処理に必要なデータを効率的に取得できることを確認した。

In this paper, we address a problem of MaxRS (Maximizing Range Sum) query processing in mobile ad hoc networks (MANETs). We assume that a MANET consists of mobile nodes which hold spatial data items with scores. A query issuer specifies a size of a rectangle and issues a MaxRS query. Then, it retrieves the location of the rectangle which maximizes the sum of the scores of all data items covered by the rectangle. For instance, to conduct rescue operations efficiently, rescue workers retrieve a dense location where many injured people exist by issuing a MaxRS query. Although a query issuer can calculate an answer to a MaxRS query by retrieving all data items in a network, this approach is inefficient since the network

bandwidth is limited in MANETs. Therefore, we propose an efficient method for MaxRS query processing in MANETs. In our method, a query issuer limits an area including necessary data items for calculating an answer to a MaxRS query in phase 1. In phase 2, it retrieves only necessary data items which exist in the limited area, which results in reducing data transmissions. The result of experiments demonstrated that our method efficiently retrieves necessary data items for MaxRS query processing.

1. はじめに

近年、IEEE802.11 や Bluetooth, Wi-Fi Direct などの無線通信技術の発展や、スマートフォンおよびタブレット端末などの計算機の小型化や高性能化に伴い、無線通信端末のみで一時的に構成されるアドホックネットワークへの関心が高まっている [8]。アドホックネットワークは既存の通信インフラが必要でないため、災害地などのインターネットが利用できない環境への応用が期待されている [1]。

通信インフラを利用できない災害地において、アドホックネットワークを用いて、救急隊員が救護活動を行うことを考える。救急隊員は、それぞれ無線通信端末を保持し、それらの端末によりアドホックネットワークが構成されている。また、救急隊員は救護活動中に取得した負傷者情報を自身の端末に保持する。このような状況において救護活動を効率的に行うために、ある救急隊員が、最も救護が必要とされている場所を検索することが考えられる。このような要求を満たすものとして、MaxRS (Maximizing Range Sum) 検索が存在する [5]。MaxRS 検索は、クエリ発行端末がある大きさの長方形を指定し、その長方形に包含される位置データのスコアの和が最大となる長方形の位置を取得するものである。上記に示した例を考えると、データの位置を負傷者の位置、データのスコアを負傷者の負傷度としてMaxRS 検索を実行することにより、負傷者の負傷度の和が最大となる位置、つまり、最も救護が必要と考えられる場所を検索できる。以下の例を用いて、MaxRS 検索を説明する。

例 1.1. 図 1 は MaxRS 検索の一例を表している。最も外側の太線の長方形はネットワーク領域、ネットワーク領域内の破線および実線の長方形はクエリ発行端末が指定した大きさの長方形、黒点はスコアをもつ位置データを表している。簡単化のため、位置データのスコアは全て 1 とする。この例では、最も多くの黒点を包含している実線の長方形が MaxRS 検索の解の 1 つである。

MaxRS 検索は、例 1.2 のようなアプリケーションに適応することも考えられる。

例 1.2. アドホックネットワークの一種であるモバイルセンサネットワークを考える。ネットワークを構成するセンサ端末が一秒毎に位置情報を記録して、その情報を保持しているものとする。例え

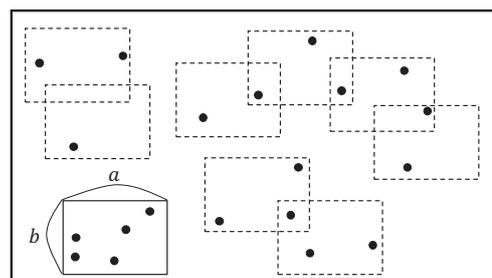


図 1: MaxRS 検索の一例

♡ 学生会員 大阪大学大学院情報科学研究科
nakayama.yuki@ist.osaka-u.ac.jp

◇ 正会員 大阪大学大学院情報科学研究科
{amagata.daichi, hara}@ist.osaka-u.ac.jp

♣ 正会員 大阪大学
nishio@ist.osaka-u.ac.jp

ば、それぞれの端末内で保持したある一定時間の位置情報データをデータ集合として MaxRS 検索を実行すると、ある一定時間中にセンサ端末が最も集中して存在していた場所を特定できる。モバイルセンサネットワークは、様々な位置のセンサ値を取得したいといった要求があり、上記の MaxRS 検索を実行することにより、センサ端末が密集している場所を把握し、センサ端末の位置を分散させることができる。

上記のように実用的なアプリケーションが存在するにも関わらず、アドホックネットワークにおける MaxRS 検索はこれまでに研究されていない。そこで本稿では、アドホックネットワークにおける効率的な MaxRS 検索手法を提案する。提案手法は 2 フェーズによる MaxRS 検索を行う。1 フェーズ目では、MaxRS 検索の解となり得る領域を限定する。さらに、2 フェーズ目で、1 フェーズ目で特定した領域に存在するデータを取得し、クエリ発行端末が正確な MaxRS 検索の解を計算する。このように、不要なデータの送信を抑制することにより、効率的に MaxRS 検索を実行できる。シミュレーション実験により、提案手法は取得精度を高く維持しつつ、トラヒックを削減できることを確認した。

以下では、2 章で本稿での問題定義および単純手法について説明し、3 章で提案手法について述べる。4 章でシミュレーション実験により、提案手法の性能を評価する。5 章で関連研究について述べ、最後に、6 章で本稿をまとめる。

2. 事前準備

本章では、まず、想定環境および MaxRS クエリを定義する。その後、アドホックネットワークにおける MaxRS クエリ処理の単純な手法について述べる。

2.1 問題定義

想定環境. n 台の端末 (識別子: M_1, M_2, \dots, M_n) でアドホックネットワークが構成され、各々が自由に移動する。各端末は、端末が移動する最大範囲であるネットワーク領域を把握しているものとする。また、各端末 M_i が保持しているデータ集合を O_i と表す。あるデータ $o \in O_i$ は、二次元の位置情報 $(o.x, o.y)$ およびスコア ($o.score$) を保持している。また、スコアは何らかのスコアリング関数により計算される 0 より大きい値とする。ネットワーク中の全データ集合 $O (= \cup O_i)$ において、複製は存在しないものとする。

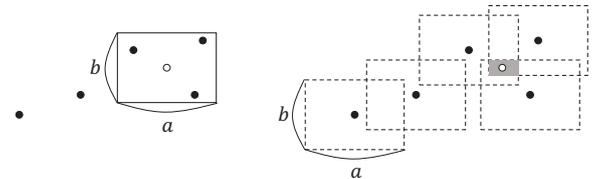
定義 1 (MaxRS 問題). ネットワーク領域内の無限地点集合 P 、およびネットワーク中の全てのデータの集合 O が与えられる。また、大きさが指定された長方形の中心を $p \in P$ とする。このとき、 $\sum_{o \in O_{R(p)}} o.score$ が最大となる長方形の中心 p を MaxRS 問題の解とする。ただし、 $R(p)$ は地点 p を中心とする指定された大きさをもつ長方形、 $O_{R(p)}$ は長方形 $R(p)$ 内に存在するデータ集合である。

ここで、地点集合 P の要素数が無限であるため、MaxRS 問題を正確に解くためには、無限回のスコアの和の計算が必要となり、非現実的である。そこで、MaxRS 問題を次のように変換する。

定義 2 (Rectangle intersection 問題 (RI 問題)). 二次元平面上に存在している長方形の集合 R が与えられる。このとき、最も多くの長方形が交差している領域を RI 問題の解とする。

文献 [9] において、全てのデータのスコアが 1 である場合の MaxRS 問題と MaxRS 問題におけるデータの位置を長方形の中心とした長方形の集合を入力とする RI 問題が等価であることが示されている。図 2 を用いて、これを説明する。

図 2(a) において、黒点はデータの位置、白点は横および縦の長さがそれぞれ a および b の長方形が指定された場合の MaxRS 問題の解の 1 つ、長方形は白点を中心とする横および縦の長さがそれぞれ a および b の長方形を表している。このとき、データの位置を中心とする横および縦の長さがそれぞれ a および b の長方形



(a) MaxRS 問題 ($\forall o \in O$ について, $o.score = 1$)

(b) RI 問題

図 2: MaxRS 問題の変換例

の集合が与えられたときの RI 問題を考える (図 2(b)). この RI 問題の解は、図 2(b) における灰色の領域である。図 2 に示す通り、MaxRS 問題の解の 1 つである白点は、RI 問題の解である灰色の領域中に存在する。このように、全てのデータのスコアが 1 である場合の MaxRS 問題を RI 問題に変換し、RI 問題を解くことにより、元の問題の解が得られる。さらに、RI 問題は、文献 [7] において、平面走査法をもとにしたアルゴリズムが提案されており、 $O(n \log n)$ の計算量で解くことができる (n は RI 問題で与えられる長方形の数)。ここまで、全てのデータのスコアが 1 である場合を考えていたが、本稿では、データのスコアを 1 と限定しない。よって、RI 問題を拡張するために、以下の定義を導入する。

定義 3 (領域のスコア). ネットワーク中の全てのデータの集合 O および大きさが指定された長方形が与えられたとき、それらのデータの位置を中心とする指定された大きさをもつ長方形の集合 R が得られる。また、 $o.score$ を対応する長方形のスコアとする。このとき、ある領域 s のスコアは、 s に重なっている長方形のスコアの和である。

長方形のスコアを考慮し、文献 [7] で提案されているアルゴリズムを利用することにより、スコアが最も大きい領域を求めることができる。つまり、0 より大きい値のスコアをもつ n 個の位置データのデータ集合が与えられた MaxRS 問題は、 $O(n \log n)$ の計算量で解ける [5], [6]。以下に、本稿で考えるクエリを定義する。

定義 4 (MaxRS クエリ). ネットワーク中の全てのデータの集合 O および大きさが指定された長方形が与えられたとき、ネットワーク領域中でスコアが最も大きい領域を求める。

本稿では、クエリ発行端末が、長方形の横の長さ a 、縦の長さ b を指定し、上記の MaxRS クエリの解を取得する問題について考える。また、指定した長方形の回転 [3] は考えない。

2.2 単純手法

MaxRS クエリの解を正確に取得する単純な手法として、クエリ発行端末が、ネットワーク内の全てのデータを収集し、収集したデータをもとに MaxRS 問題を解くことが考えられる。以下に、その詳細を記す。

まず、クエリ発行端末がネットワーク内の全ての端末にデータ要求メッセージをフラッディングする。このメッセージには、送信元端末 ID、親端末 ID、およびホップ数 (初期値は 0) が添付されている。データ要求メッセージを初めて受信した端末は、メッセージ中の送信元端末を自身の親端末とし、ホップ数を 1 増やしたデータ要求メッセージをブロードキャストする。さらに、この端末は親端末を送信先とする ACK メッセージを送信する。また、この端末はデータ返信を行うまでの時間 (データ返信待ち時間 WT) を以下のように設定する。

$$WT = Delay_{max} \cdot \frac{HOP_{max} - HOP_{count}}{HOP_{max}}$$

ここで、 $Delay_{max}$ は最大待ち時間、 HOP_{max} はネットワーク中の最大ホップ数、 HOP_{count} はクエリ発行端末までのホップ数である。

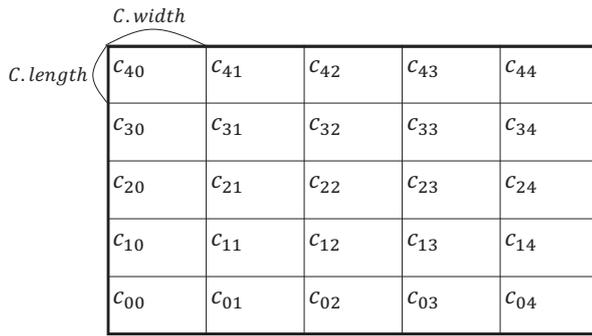


図 3: ネットワーク領域の分割

また、ある端末がデータ要求メッセージを受信した時にメッセージ中の親端末 ID が自身の端末 ID であった場合、または自身を送信先とする ACK メッセージを受信した場合、そのメッセージ中の送信元端末を自身の子端末とする。その後、親端末 ID が自身の端末 ID であるメッセージを受信しなかった端末は、自身を葉端末とみなし、自身の保持する全てのデータを親端末へ返信する。データを受信した端末は、送信元端末を送信先とする ACK メッセージを送信する。自身を送信先とする ACK メッセージを受信した端末は、データを親端末に送信できたとみなし、これ以降なにもしない。一方、一定時間待った後、自身を送信先とする ACK メッセージを受信していない端末は、データを親端末に送信できていないとみなし、データを親端末へ再送信する。自身の全ての子端末からデータを受信した端末、およびデータ返信待ち時間が経過した端末は、自身の保持する全てのデータおよび子端末から受信した全てのデータを親端末へ返信する。これにより、クエリ発行端末はネットワーク内の全てのデータを収集できる。その後、クエリ発行端末は、文献 [7] で提案されているアルゴリズムを用いて、収集したデータおよび指定した長方形 (a, b) から MaxRS クエリの解を求める。

このように、単純手法は、クエリ発行端末がネットワーク内のデータを全て収集する。したがって、ネットワーク内のデータ数が多い場合、トラフィックが大きくなり、パケット損失や通信遅延が起きる可能性が高くなるため、効率的ではない。

3. 提案手法

提案手法は、2 フェーズで MaxRS 検索を実行する。初めに、1 フェーズ目において、クエリ発行端末は、MaxRS クエリの解となり得る領域を求める。そのために、ネットワーク領域をグリッド状に分割し、分割したセルの各辺の長さ $(C.width, C.length)$ をクエリに添付する。セルの各辺の長さはそれぞれ $C.width = \frac{a}{k}$, $C.length = \frac{b}{l}$ (k, l は自然数) とする。その後、クエリをネットワーク全体にフラッディングする。このとき、単純手法と同様の方法で、クエリ発行端末を根とする木構造のネットワークトポロジを構築する。クエリを受信した端末は、クエリに添付されているセルの各辺の長さを用いて、クエリ発行端末と同様にネットワーク領域をグリッド状に分割する。ネットワーク領域をグリッド上に分割した例を図 3 に示す。この例では、ネットワーク領域を 5×5 の合計 25 個のセル (セル c_{00} ~ セル c_{44}) に分割している。その後、セルの数分のスコアを格納できるサイズの配列を返信メッセージに添付し、葉端末から順にセルのスコアの和を親端末に送信する。全ての端末でネットワーク領域の分割方法を共有しているため、セルと配列の格納位置の対応関係は全ての端末で共通である。

以下に、1 フェーズ目における処理について説明する。まず、葉端末は自身のもつデータの位置に対応するセルにそのデータのスコアを加算する。対応するデータが存在しないセルのスコアは 0 とする。このように作成した配列を返信メッセージに添付し、親端末に送信する。返信メッセージを受信した端末は、単純手法と同様

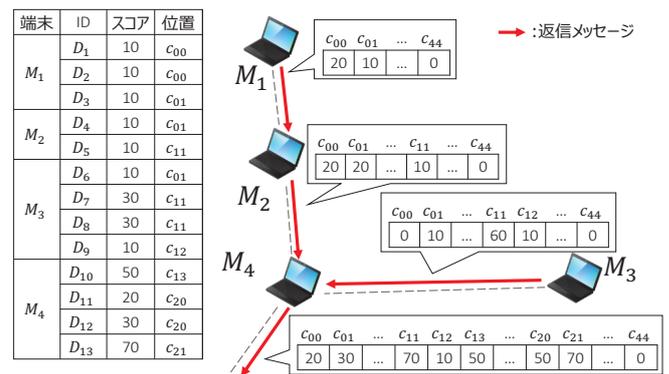


図 4: 1 フェーズ目の返信例

に ACK メッセージを子端末へ送信する。この ACK メッセージについては、単純手法のデータ返信時と同様の処理を行う。さらに、この端末は、葉端末と同様に、返信メッセージ中のセル毎のスコアを格納した配列中の、自身のもつデータの位置に対応するセルに、そのデータのスコアを加算し、親端末へ返信メッセージを送信する。複数の子端末から返信メッセージを受信した場合、全ての返信メッセージ中のセル毎のスコアの和をとったものに、自身のもつデータのスコアを加算する。これを、例を用いて説明する。

例 3.1. 図 4 は 1 フェーズ目の返信の例を示している。ネットワーク領域の分割は図 3 に従い、返信メッセージには 25 個のスコアを格納できるサイズの配列を用いる。図 4 における左の表は、データとそれを所持する端末の対応関係、吹出しの長方形は返信メッセージ中の配列の中身の一部を表している。簡単化のため、データの位置をセル ID で表記しているが、実際は、緯度および経度の二次元の数値で表される。葉端末 M_1 は、データ D_1, D_2 および D_3 の 3 つのデータを所持しており、それぞれの位置はセル c_{00}, c_{00} および c_{01} に対応する。したがって、返信メッセージ中の配列を $c_{00} = 10 + 10 = 20$, $c_{01} = 20$ とし (他のセルの値は 0)、親端末 M_2 に送信する。また、もう一台の葉端末 M_3 も M_1 と同様の方法により、返信メッセージを親端末 M_4 に送信する。 M_1 からのメッセージを受信した M_2 は、自身のもつデータのスコアを返信メッセージ中の配列の対応する箇所に加算する。つまり、 $c_{00} = 20$, $c_{01} = 10 + 10 = 20$, $c_{11} = 10$ とする。 M_2 は、このようにして更新したメッセージを親端末 M_4 に送信する。 M_4 は、二台の子端末 M_2 および M_3 からメッセージを受信したとき、返信メッセージ中の配列の対応する箇所の和をとり、さらに自身のもつデータのスコアを対応する箇所に加算する。すなわち、 $c_{00} = 20$, $c_{01} = 20 + 10 = 30$, $c_{11} = 60 + 10 = 70$, $c_{12} = 10$, $c_{13} = 50$, $c_{20} = 20 + 30 = 50$, $c_{21} = 70$ とする。その後、 M_4 も同様に、更新したメッセージを親端末に送信する。

これにより、1 フェーズ目において、クエリ発行端末はネットワーク領域全体のセル毎のスコアの和を得る。次に、クエリ発行端末はセル毎のスコアの和から、正確な MaxRS クエリの解を求めるために必要であるデータが存在するセルの集合 (データ要求セル集合) を計算する。ここで、ある任意のセルを c_{ij} とし、 c_{ij} の 1 つ右のセルおよび 1 つ上のセルをそれぞれ $c_{i(j+1)}$ および $c_{(i+1)j}$ とする。このとき、セル集合 C_{ij}^{kl} を $C_{ij}^{kl} = \cup_{j \leq x < j+k} \cup_{i \leq y < i+l} C_{yx}$, セル集合 C_{ij}^{kl} に含まれるセルのスコアの和を $score(C_{ij}^{kl})$ と定義する。ただし、ネットワーク領域外のセルのスコアは 0 とみなす。また、ネットワーク領域を分割した全てのセルの集合を C_{all} と表す。まず、クエリ発行端末はネットワーク領域を分割した全てのセル $c_{ij} \in C_{all}$ に対して $score(C_{ij}^{kl})$ を計算する。 $a = C.width \times k$ および $b = C.length \times l$ であり、 C_{ij}^{kl} の大きさがクエリ発行端末の指定した長方形の大きさと等しいため、 $\max_{c_{ij} \in C_{all}} \{score(C_{ij}^{kl})\}$ は MaxRS クエリの解となる領域のスコアの下限值となる。これを lb_{score} と表す。次に、クエ

c_{40}	c_{41}	c_{42}	c_{43}	c_{44}
20	30	40	10	0
c_{30}	c_{31}	c_{32}	c_{33}	c_{34}
70	10	20	0	20
c_{20}	c_{21}	c_{22}	c_{23}	c_{24}
90	100	50	20	20
c_{10}	c_{11}	c_{12}	c_{13}	c_{14}
270	300	40	30	20
c_{00}	c_{01}	c_{02}	c_{03}	c_{04}
100	80	50	30	0

$score(C_{03}^{22})=80$

図 5: セル毎のスコアの和の例

リ発行端末はネットワーク領域を分割した全てのセル $c_{ij} \in C_{all}$ に対して $score(C_{ij}^{(k+1)(l+1)})$ を計算する. $score(C_{ij}^{(k+1)(l+1)}) \geq lb_{score}$ となるセル集合 $C_{ij}^{(k+1)(l+1)}$ に含まれるセルは, MaxRS クエリの解を求めるために必要であるデータが存在する可能性があるため, データ要求セル集合の要素とする. これにより, クエリ発行端末は, 2 フェーズ目においてデータを要求しなければならない領域を把握できる. 以下の例を用いて, $k = l = 1$ の場合の計算方法を説明する.

例 3.2. 図 5 は, クエリ発行端末が取得したセル毎のスコアの和の例を示している. セル内の長方形に囲まれた数値は, セルのスコアを表している. 例えば, セル c_{00} のスコアは 100, セル c_{01} のスコアは 80 である. $k = l = 1$ のため, $lb_{score} = \max_{c_{ij} \in C_{all}} \{score(C_{ij}^{11})\} = 300$ (セル c_{11}) である. 次に, ネットワーク領域を分割した全てのセル $c_{ij} \in C_{all}$ に対して $score(C_{ij}^{22})$ を計算する. 例えば, セル c_{03} , c_{04} , c_{13} , および c_{14} のスコアの和 $score(C_{03}^{22})$ は 80 である. これは, lb_{score} 未満であるため, この時点ではセル c_{03} , c_{04} , c_{13} , および c_{14} はデータ要求セル集合の要素とはならない. 全てのセルに対して計算を行ったとき, lb_{score} 以上のスコアとなるセル集合は, $\{c_{00}, c_{01}, c_{10}, c_{11}\}$, $\{c_{01}, c_{02}, c_{11}, c_{12}\}$, $\{c_{10}, c_{11}, c_{20}, c_{21}\}$, および $\{c_{11}, c_{12}, c_{21}, c_{22}\}$ であり, それぞれのセル集合に対応するスコアの和は 750, 470, 760, 490 である. したがって, データ要求セル集合は $\{c_{00}, c_{01}, c_{02}, c_{10}, c_{11}, c_{12}, c_{20}, c_{21}, c_{22}\}$ となり, 図 5 における灰色の領域である.

2 フェーズ目では, クエリ発行端末は, 1 フェーズ目で計算したデータ要求セル集合に含まれるセル ID をデータ要求メッセージに添付し, 1 フェーズ目と同様の方法で, データ要求メッセージをフラッディングする. 以下, データ要求セル集合に含まれるセルの領域の和集合をデータ要求領域と呼ぶ. 葉端末は, データ要求領域に存在するデータを所持している場合, そのデータを親端末へ返信する. データを受信した端末は送信元端末を送信先とする ACK メッセージを送信する. この ACK メッセージは単純手法のデータ返信時と同様の処理を行う. また, 全ての子端末からデータを受信した端末は, 子端末から受信したデータおよびデータ要求領域に存在する自身のデータを親端末へ送信する. 一方, ある端末がデータ要求領域に存在するデータを所持していない場合, データを添付せずに親端末にメッセージを送信する. また, 1 フェーズ目のメッセージを受信していない端末が 2 フェーズ目のメッセージを受信した場合, この端末はデータ要求領域を把握できないため, 自身のデータを全て親端末へ送信する. ここで, 親端末が 1 フェーズ目のメッセージを受信していた場合, データ要求領域を把握しているため, 子端末から送信されたデータのうちデータ要求領域に含まれないデータは, これ以降送信しない. 以下の例を用いて, 2 フェーズ目の返信を説明する.

例 3.3. 図 6 は, 2 フェーズ目の返信の例を示している. まず, 葉端末である M_1 および M_3 がそれぞれ親端末へメッセージを送信する. このとき, M_1 はデータ要求領域に存在するデータ (D_1) を所

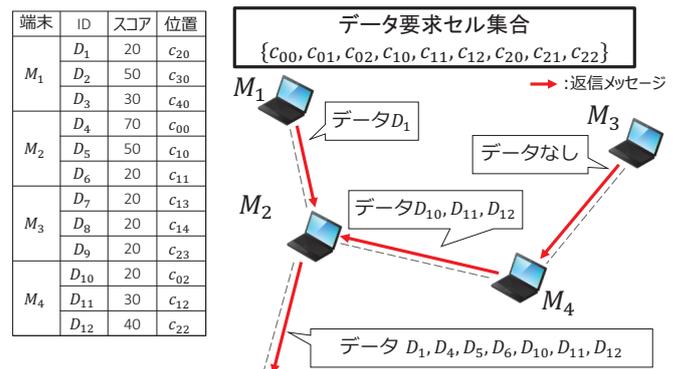
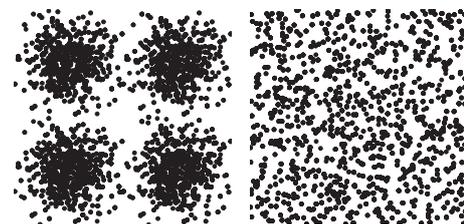


図 6: 2 フェーズ目の返信例



(a) クラスタ分布 (b) 一様分布

図 7: データの位置分布

持っているため, 親端末 M_2 にデータ D_1 を送信する. 一方, M_3 はデータ要求領域に存在するデータを所持していないため, データを添付せずに親端末 M_4 にメッセージを送信する. その後, 子端末 M_3 から返信メッセージを受信した M_4 は, データ要求領域に存在する自身のデータ D_{10} , D_{11} , および D_{12} を親端末 M_2 に送信する. 全ての子端末から返信メッセージを受信した M_2 は, 子端末から受信したデータ D_1 , D_{10} , D_{11} , および D_{12} とデータ要求領域に存在する自身のデータ D_4 , D_5 , および D_6 を親端末に送信する.

これにより, クエリ発行端末はデータ要求領域に存在するデータ, つまり, MaxRS クエリの解を求めるために必要なデータを収集できる. その後, クエリ発行端末は, 収集したデータおよび指定した長方形 (横の長さ a , 縦の長さ b) から MaxRS クエリの解を求める.

4. シミュレーション評価

本章では, 提案手法の性能評価のために行ったシミュレーション実験の結果を示す. 本実験では, ネットワークシミュレータ Qualnet6.1¹ を用いた. また, MaxRS 問題を解くアルゴリズムは, C++ で実装されており, 実験は 3.4GHz Core i7 および 8GB RAM で構成される PC 上で行った.

4.1 シミュレーション環境

600[m]×600[m] の二次元平面上の領域に n 台の端末が存在する. 各端末はランダムウェイポイントモデル [2] に従い, $[0,1.0]$ [m/s] の速度で移動し, 停止時間は 30[s] とした. 各端末は IEEE802.11b を使用し, 通信帯域 11[Mbps], 通信伝播距離が 100[m] 程度となる送信電力で通信する. 各端末はデータ ID, スコア, および二次元座標 (x, y) で構成されるデータを d 個保持するものとする. 検索開始時におけるネットワーク全体のデータの位置分布およびスコア分布を以下で説明する. データの位置分布はクラスタ分布および一様分布を用いた (図 7). クラスタ分布は, クラスタの中心

¹http://www.scalable-networks.com

の x 座標および y 座標をそれぞれ平均とする正規分布を用いて、データの位置の x 座標および y 座標をそれぞれ決定した。一様分布は、値の範囲が $[0,600]$ の中の実数値である一様乱数を用いて、 x 座標および y 座標をそれぞれ決定した。また、データのスコア分布はパレート分布の確率密度関数を基にしてデータの位置との相関性を考慮した分布（以下、パレート分布と呼ぶ）、およびスコアの範囲が $[1,1000]$ の中の整数値である一様分布を用いた。パレート分布は以下の式でスコアの計算を行った。

$$o.score = \frac{MaxScore}{2} \cdot \frac{pq^p}{w^{(p+1)}} + 20z$$

$$w = \frac{2Distance}{Distance_{max}} + 1$$

$MaxScore$ は $[200,1000]$ の一様分布により決定した各クラスタの中心地点のスコアの最大値、 $Distance$ はデータの位置とデータの位置が属するクラスタの中心地点の距離、 $Distance_{max}$ はデータの位置が属するクラスタの中心地点からの最大距離、 z は正規乱数である。 p および q はパレート分布のパラメータであり、本実験では $p = 2$ および $q = 1$ を用いた。このように計算したスコアを離散化し、 $[1,1000]$ の範囲の整数値とした。また、生成したデータをランダムに各端末に配置した。本実験において、クエリ発行端末は、 $MaxRS$ 検索における長方形を一边の長さが L である正方形で指定した。提案手法におけるセルサイズは、指定した正方形の大きさと等しいもの ($k = 1, l = 1$) とした。上で設定したパラメータを表 1 に示し、太字をデフォルト値とする。デフォルトのデータセットとして、データの位置をクラスタ分布、データのスコアをパレート分布で生成したものを使用した。

比較手法として、2.2 節で述べた単純手法を用いた。また、全ての手法において $HOP_{max} = 9$ とした。デフォルト値を用いた事前実験を行い、それぞれの手法において、取得精度の最大値を基に、最大待ち時間を決定した。具体的には、単純手法は最大待ち時間を 29[s]、提案手法は 1 フェーズ目の最大待ち時間を 3[s]、2 フェーズ目の最大待ち時間を 24[s] とした。また、提案手法 (HALF) は、2 フェーズ目の最大待ち時間を提案手法の半分とした。つまり、1 フェーズ目の最大待ち時間は 3[s]、2 フェーズ目の最大待ち時間は 12[s] とした。

以上のシミュレーション環境において、クエリ発行端末をランダムに選び、クエリを発行するという処理を 100 回繰り返す、以下の評価値の平均を調べた。

- **トラヒック**: 送信されたメッセージの総バイト数。表 2 は、各手法における全てのメッセージサイズを示す。この表において、 D はメッセージに含まれるデータ数、 $AllCell$ は分割した全てのセルの数、 $ReqCell$ はデータを要求するセルの数を表す。
- **取得精度**: $MaxRS$ 検索の正解を求めるために必要であるデータを取得した割合。
- **検索時間**: クエリ発行端末がクエリを発行してから $MaxRS$ 検索の解を取得するまでの時間。

4.2 データセットの影響

データの位置分布およびスコア分布を変化させた時の単純手法と提案手法の性能を調べた。その結果を図 8 に示す。横軸は用いたデータの位置分布およびスコア分布を示し、縦軸は、図 8(a) はトラヒック、図 8(b) は取得精度、および図 8(c) は検索時間を表す。ここで、図中の L_C および L_U は、データの位置分布がそれぞれクラスタ分布および一様分布であることを表し、 S_P および S_U は、データのスコア分布がそれぞれパレート分布および一様分布であることを表す。また、 $L_C + S_P$ はデータセットが、 L_C および S_P を用いて生成されたことを示す。

図 8(a) より、提案手法および提案手法 (HALF) はデータセットに関わらず、単純手法よりもトラヒックが小さいことが分かる。特

表 1: パラメータ設定

パラメータ	意味	デフォルト値 (範囲)
n [台]	端末数	80 (50-120)
d [個]	1 台の端末が持つデータ数	80 (1-160)
L [m]	正方形の一边の長さ	40 (20-100)

表 2: 各メッセージのサイズ

手法	メッセージの種類	サイズ [B]
単純手法	データ要求メッセージ	16
	データ返信メッセージ	$24 + 16 \times D$
提案手法	クエリメッセージ	24
	フェーズ 1 返信メッセージ	$16 + 4 \times AllCell$
	データ要求メッセージ	$24 + 4 \times ReqCell$
	データ返信メッセージ	$24 + 16 \times D$
	フェーズ 1 返信メッセージの ACK	12
両手法	メッセージフラッディング時の ACK	12
	データ返信メッセージの ACK	16

に、スコア分布がパレート分布の場合、提案手法および提案手法 (HALF) は、単純手法よりも大幅にトラヒックが小さい。具体的には、提案手法は、位置分布がクラスタ分布の時は単純手法の約 50% のトラヒック、データが一様分布の時は単純手法の約 35% のトラヒックである。これは、スコア分布がパレート分布の場合、提案手法は 1 フェーズ目でデータ要求を行うセルを効率的に限定でき、2 フェーズ目において、返信するデータ数が少なくなるためである。

図 8(b) は、データセットに関わらず、提案手法が単純手法よりも取得精度が高いことを示している。このことから、提案手法は 1 フェーズ目において、 $MaxRS$ 検索の計算に必要なデータが存在する領域を高い精度で特定できていることが分かる。

図 8(c) より、単純手法において、データセットは、検索時間にあまり影響しないことが分かる。一方、提案手法および提案手法 (HALF) は、位置分布がクラスタ分布およびスコア分布がパレート分布の時に、検索時間が最も小さい。これは、上記のデータセットの場合、2 フェーズ目に収集すべきデータ数が最も少なくなるため、2 フェーズ目においてパケット損失が起きる可能性が低くなるからである。

4.3 データ数の影響

データ数 d を変化させた時の提案手法の性能を調べた。その結果を図 9 に示す。横軸はデータ数 d を示し、縦軸は、図 9(a) はトラヒック、図 9(b) は取得精度、および図 9(c) は検索時間を表す。

図 9(a) および図 9(b) の結果より、全ての手法において、 d が大きくなるとトラヒックが増加し、取得精度が低下することが分かる。これは、ネットワーク内に存在するデータ数が多くなり、全ての手法において、返信するデータ数が多くなるためである。また、 d が大きい場合、提案手法および提案手法 (HALF) は、単純手法と比較してトラヒックが小さく、提案手法は、 d に関わらず、高い取得精度を維持していることが分かる。これは、提案手法において、1 フェーズ目でデータ要求を行うセルを限定することにより、2 フェーズ目で返信するデータ数を抑制できていることを示している。一方、提案手法 (HALF) は、提案手法よりも取得精度が低くなっている。これは、2 フェーズ目の最大待ち時間が短いため、同時にメッセージを送信する端末が多くなり、パケット損失が起こる可能性が高くなるからである。

図 9(c) より、全ての手法において、 d が大きくなると検索時間が長くなっていき、最大待ち時間に収束することが分かる。全ての手法において、 d が大きくなると、送信するデータ数が多くなり、パケット損失が増加する。これにより、中継端末が、子端末からデータを受信できず、データ返信待ち時間が経過するまで待機する可能性が高くなるためである。しかし、提案手法は、他の手法に比べて、検索時間が最大待ち時間に収束する d の値が大きい。

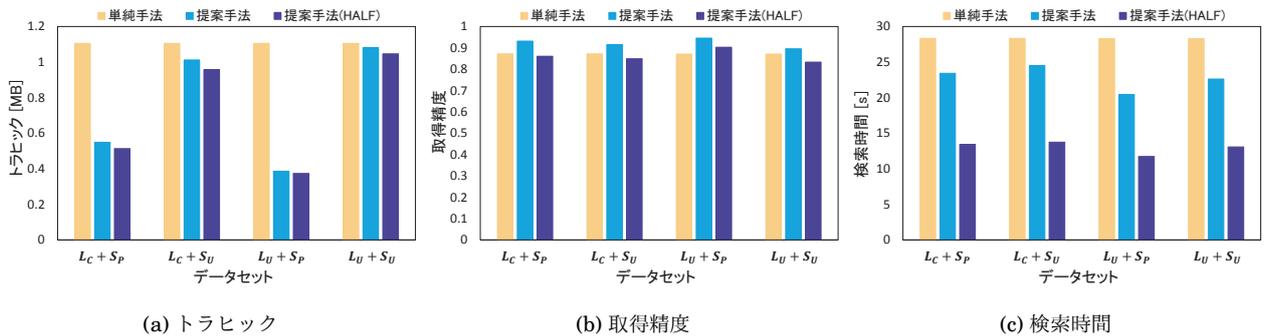


図 8: データセットによる影響

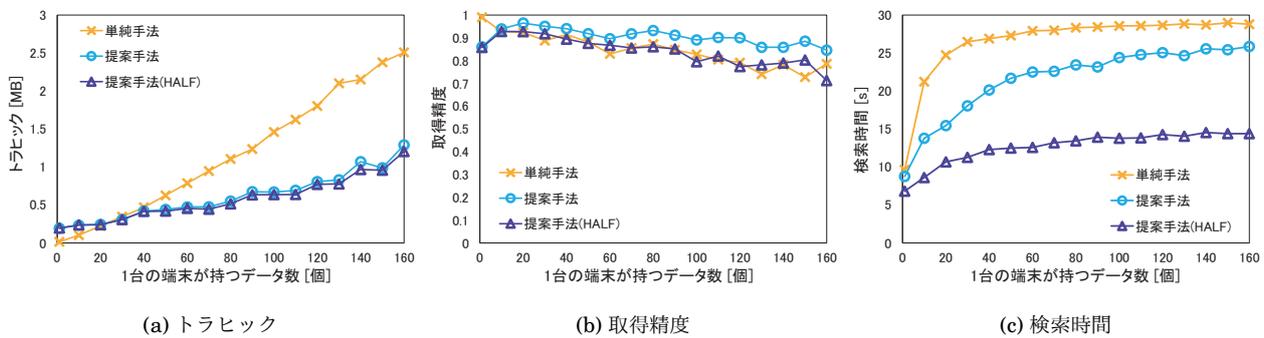


図 9: データ数 d の影響

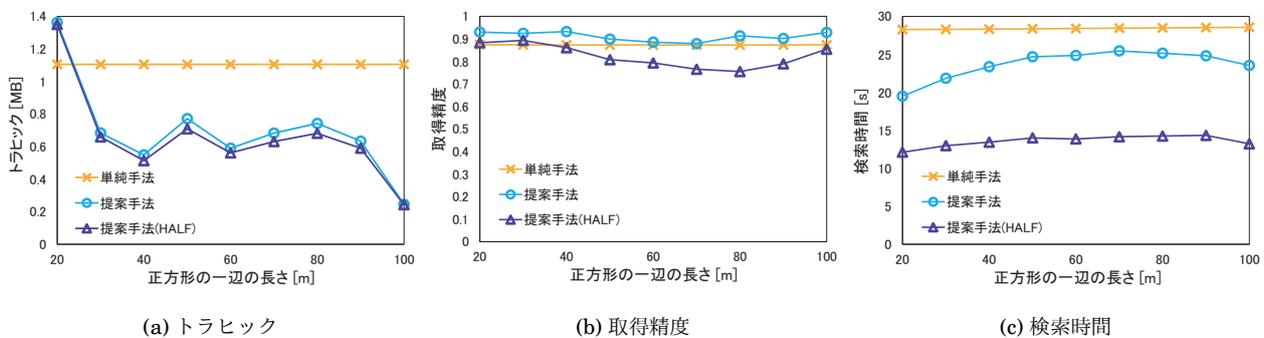


図 10: 正方形の一辺の長さ L の影響

4.4 正方形の一辺の長さの影響

正方形の一辺の長さ L を変化した時の提案手法の性能を調べた。その結果を図 10 に示す。横軸は正方形の一辺の長さ L を示し、縦軸は、図 10(a) はトラフィック、図 10(b) は取得精度、および図 10(c) は検索時間を表す。

図 10(a) の結果より、単純手法は L の影響を受けずトラフィックが一定であるのに対し、提案手法および単純手法 (HALF) は、 L の影響を受けてトラフィックが変化していることが分かる。これは、単純手法は L の大きさに関わらず、ネットワーク中の全てのデータを収集するためである。一方、提案手法および提案手法 (HALF) は $L = 30$ から $L = 90$ の時にトラフィックがほぼ変化せず、 $L = 100$ の時にトラフィックが小さくなっていることが分かる。 L が小さくなると、ネットワーク領域を分割した際のセルの数が増加するため、提案手法における 1 フェーズ目のメッセージサイズが大きくなる。また、 $L = 30$ から $L = 90$ の時、分割した全てのセルの数に対するデータ要求セルの数の割合が大きくなり、2 フェーズ目

において返信するデータ数が多くなる。

図 10(b) より、単純手法は L の影響を受けず取得精度が一定であるのに対し、提案手法および提案手法 (HALF) は、 L の影響を受けて取得精度が変化していることが分かる。これは、上に述べたように、単純手法が、 L の大きさに依存しない手法であるためである。また、提案手法は全ての場合において、単純手法よりも取得精度が高い。これは、提案手法が 2 フェーズの通信を行うことで、返信するデータ数を削減しているためである。また、提案手法 (HALF) は L が 40 以上のとき、単純手法より取得精度が小さい。これは、4.3 節で述べたように、最大待ち時間が短く、同時にメッセージを送信する端末が多くなり、パケット損失が多くなるためである。

図 10(c) の結果より、単純手法は L の影響を受けず検索時間がほぼ一定であるのに対し、提案手法および提案手法 (HALF) は、 L の影響を受けて検索時間が変化している。2 フェーズ目で取得する必要のあるデータ数が L によって変わるため、それらのデー

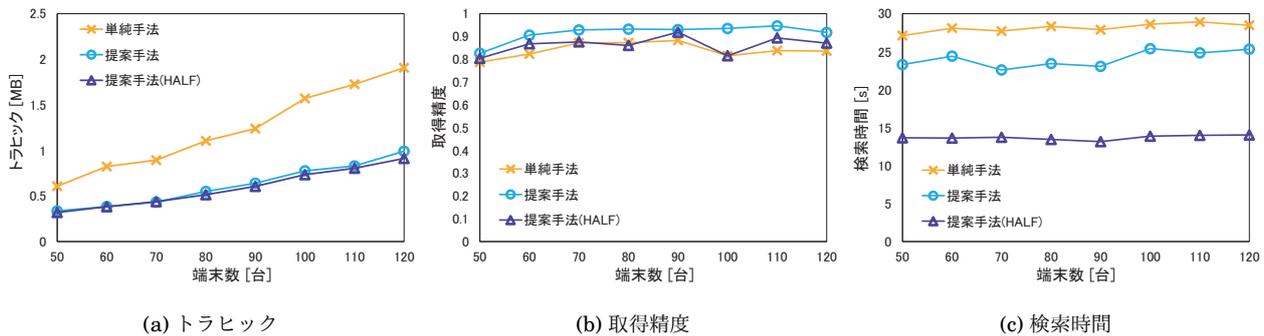


図 11: 端末数 n の影響

タを全て受信するまでの時間も変わるからである。

4.5 端末数の影響

端末数 n を変化した時の提案手法の性能を調べた。その結果を図 11 に示す。横軸は端末数 n を示し、縦軸は、図 11(a) はトラフィック、図 11(b) は取得精度、および図 11(c) は検索時間を表す。

図 11(a) より、全ての手法において、端末数が増えるほどトラフィックが増加することが分かる。これは、端末数が増えるほど、ネットワーク内のデータ数が増え、返信するデータ数が増えるためである。また、図 11(a) および図 11(b) より、端末数に関わらず、提案手法および提案手法 (HALF) は、単純手法よりもトラフィックが小さく、ほとんどの場合において、単純手法よりも取得精度が高いことが分かる。これは、提案手法および提案手法 (HALF) において、1 フェーズ目でデータ要求セルを限定することが、取得精度を維持しつつトラフィックを削減する上で有効であることを示している。

図 11(c) より、全ての手法において、端末数に関わらず、検索時間がほぼ一定であることが分かる。これは、端末数により、平均ホップ数が増えるためである。

5. 関連研究

集中管理データベースの分野において、MaxRS 検索を効率的に行う様々な手法が提案されている [3], [5], [6], [9], [10]. 文献 [9] では、全てのデータをメモリ上に格納できる環境において、文献 [5], [6] では、全てのデータをメモリ上に格納できない環境において、正確な MaxRS 検索を実行する手法をそれぞれ提案している。また、文献 [10] では、全てのデータをメモリ上に格納できない環境における近似アルゴリズムを提案している。文献 [3] では、指定された長方形の回転を考慮したアルゴリズムを提案している。これらの研究は、集中管理データベースにおいて、計算量の削減や、ディスク I/O の削減を目的としており、分散環境であるアドホックネットワークに適応することは困難である。

無線センサネットワークの分野において、領域検索問題を扱う研究が存在する [4], [11]. 文献 [11] では、消費電力の抑制を目的とした max regional aggregate (MaxRA) クエリ処理手法が提案されている。MaxRA クエリは、領域の大きさを指定し、指定された大きさの領域内に含まれるセンサ端末のセンサ値の集約値 (平均値や合計値) が最も大きい領域を検索するものである。つまり、MaxRS クエリは MaxRA クエリの一部といえる。この文献では、精度を保証した上で、データを取得する領域およびセンサ端末をサンプリングする近似手法を用いている。また、文献 [4] では、region-based クエリ処理手法が提案されている。region-based クエリは、値の範囲および領域の大きさを指定し、指定された大きさの領域内に含まれるセンサ端末のセンサ値の集約値が、指定された値の範囲に含まれる領域を検索するものである。この文献では、クラスタリングベースのクエリ処理手法を用いて、消費電力

を抑制している。具体的には、基地局が、集約すべき領域を全て計算し、それに応じたクラスタヘッドの選択の最適化問題を解いている。これらの研究は、条件を満たす領域を検索するという点において、本研究における MaxRS クエリと同様であるが、本稿で扱う問題とは異なるため、これらの研究で提案されている手法は適用できない。

6. おわりに

本稿では、アドホックネットワークにおいて、高い取得精度を維持しつつトラフィックを削減することを目的とする MaxRS 検索処理手法を提案した。提案手法では、1 フェーズ目に、ネットワーク領域をグリッド状に分割し、分割したセル毎のスコアを収集する。さらに、収集した情報をもとにデータを収集する必要がある領域を限定し、2 フェーズ目で、特定した領域に存在するデータのみを収集する。これにより、MaxRS 検索の解の計算に不必要なデータの送信を抑制できる。シミュレーション実験の結果から、提案手法が高い取得精度を維持しつつ、トラフィックを削減できていることを確認した。

ネットワーク内のデータ数が多い場合、データを収集する必要がある領域を限定した場合でも、2 フェーズ目で返信するデータ数が増え、取得精度が低下してしまう。また、提案手法はデータのスコアの分布により、トラフィックおよび取得精度が大きく影響される。そこで、返信メッセージのトラフィックをさらに抑制する手法を設計することを検討している。

【謝辞】

本研究の一部は、文部科学省研究費補助金・基盤研究 (A)(JP26240013) および JST 国際科学技術共同研究推進事業 (戦略的国際共同研究プログラム) 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. Amagata, Y. Sasaki, T. Hara, and S. Nishio, "CTR: An Efficient Cluster-based Top-k Query Routing in MANETs", In *MoMM*, pp.225-234, 2014.
- [2] T. Camp, J. Boleng, and V. Davies, "A survey of mobility models for ad hoc network research", *Wireless Communications and Mobile Computing*, 2(5), pp.483-502, 2002.
- [3] Z. Chen, Y. Liu, R. C.-W. Wong, J. Xiong, X. Cheng, and P. Chen, "Rotating MaxRS queries", *Information Sciences*, 305, pp.110-129, 2015.
- [4] D.-W. Choi and C.-W. Chung, "REQUEST+: A framework for efficient processing of region-based queries in sensor networks", *Information Sciences*, 248, pp.151-167, 2013.

- [5] D.-W. Choi, C.-W. Chung, and Y. Tao, "A scalable algorithm for maximizing range sum in spatial databases", *PVLDB*, 5(11), pp.1088-1099, 2012.
- [6] D.-W. Choi, C.-W. Chung, and Y. Tao, "Maximizing Range Sum in External Memory", *ACM TODS*, 39(3), p. 21, 2014.
- [7] H. Imai and T. Asano, "Finding the connected components and a maximum clique of an intersection graph of rectangles in the plane", *Journal of algorithms*, 4(4), pp.310-323, 1983.
- [8] Y. Komai, Y. Sasaki, T. Hara, and S. Nishio, "kNN Query Processing Methods in Mobile Ad Hoc Networks", *IEEE TMC*, 13(5), pp.1090-1103, 2014.
- [9] S.-C. Nandy and B.-B. Bhattacharya, "A unified algorithm for finding maximum and minimum object enclosing rectangles and cuboids", *Computers & Mathematics with Applications*, 29(8), pp.45-61, 1995.
- [10] Y. Tao, X. Hu, D.-W. Choi, and C.-W. Chung, "Approximate MaxRS in spatial databases", *PVLDB*, 6(13), pp.1546-1557, 2013.
- [11] Y. Zhuang and L. Chen, "Max regional aggregate over sensor networks", In *ICDE*, pp.1295-1298, 2009.

中山 侑紀 Yuki NAKAYAMA

2015年大阪大学工学部卒業。大阪大学大学院情報科学研究科在学中。

天方 大地 Daichi AMAGATA

大阪大学大学院情報科学研究科助教。2015年大阪大学大学院情報科学研究科博士後期課程修了，情報科学博士。データベースシステムの研究に従事。IEEE, ACM, 情報処理学会各会員。

原 隆浩 Takahiro HARA

大阪大学大学院情報科学研究科教授。1997年大阪大学大学院工学研究科博士前期課程修了，工学博士。データベースシステム，分散処理の研究に従事。IEEE, ACM, 電子情報通信学会各会員。

西尾 章治郎 Shojiro NISHIO

大阪大学総長。1975年京都大学工学部卒業。1980年同大学院工学研究科博士後期課程修了，工学博士。京都大学工学部助手等を経て，1992年大阪大学工学部教授，2015年大阪大学総長となり，現職に至る。文部科学省科学官，大阪大学理事・副学長等を歴任。データ工学の研究に従事。本会理事，監事，会長を歴任。紫綬褒章を受章し，本会より功労賞，論文賞を受賞。