

経路情報を用いた複数タスクへのセンサ割当て

鳥海 晋^{†1} 本位田 真一^{†2,†1}

複数のユーザが複数のタスクを投入する共有型の無線センサネットワーク (WSN) においては資源競合問題を解消する必要がある。システムは重要度の高いタスクを優先しながら多くのタスクを実行することが求められる。そのため、この問題はタスクに対しセンサを適切に割り当て、得られる利得を最大化する問題として定式化できる。既存の研究においては、単純なモデルを用いてセンシングに消費する電力のみをモデル化していたが、WSN では通信に消費する電力が重要であることが一般に知られている。上記のような背景から、本論文では、通信に関わるセンサ・タスク間割当て問題を定式化するとともに、割当てノードからベースステーションまでの通信路を考慮した割当て決定アルゴリズムを提案する。我々は対象としているネットワークがクエリに応答するシステムであることを利用して、クエリに通信路の情報を上乗せする。観測データを送る通信路を考慮することによって、既存手法と比較して、タスク実行にともなうより多くの利得を獲得できる。ソフトウェアによるシミュレーション結果より、提案した割当て手法が通信路を考慮しない既存手法と比較して得られる利得が向上したことを示した。

Sensor Assignment for Multiple Tasks Using Path Information

SUSUMU TORIUMI^{†1} and SHINICHI HONIDEN^{†2,†1}

A “shared” wireless sensor network (WSN) where multiple users deploy multiple tasks requires a mechanism for resource arbitration. The system is required to perform as many tasks as possible with prioritizing important tasks. This problem can be formulated as a maximization problem of profit gained by assigning appropriate sensors to each task. The previous approaches only dealt with energy used for sensing; however, experiments have shown that the energy used for communication must be dealt with in the context of a WSN. In this paper, we propose a novel formulation of the above problem and a distributed assignment algorithm that takes into account path information between the assigned node and the base station. We take advantage of the “reactive” nature of the network and piggyback path information on queries. By preserving routes through which sensing data run, we can expect higher profit gained in compar-

ison with previous approaches. The results of software simulations showed that our assignment algorithm can achieve higher overall profit than the previous one that does not exploit information about the path.

1. 導 入

ハードウェア技術の発達と小型化によりパーベイシブコンピューティング¹⁷⁾への関心が高まっている。パーベイシブコンピューティングを実現するためには、アプリケーションは現実環境の情報を取得する必要がある。無線センサネットワーク (WSN) はパーベイシブコンピューティング環境の基盤として考えられており、数々の WSN アプリケーションが提案されている。その中には、環境観測²⁰⁾、物体追跡¹³⁾、生態観測¹¹⁾といったものが含まれる。

さらに、複数種類のセンサを搭載したノードが実用化されており⁹⁾、こうしたノードを組み合わせることで、1つの WSN が画像データと音響データの同時取得といった複数の観測タスクを行うことができる。したがって、様々な用途に用いることができる WSN を、複数の WSN アプリケーションで共有するというシナリオが実現可能である。しかし、センサノードの資源は限られているため、各タスクに適切に資源を割り振る必要がある。センサノードは通常バッテリー駆動であり、広範囲に敷設されるため、ノードの資源のうち特に電力資源に注目することが WSN を長期間にわたって利用するために必要である。こうした背景のもと、本論文では、与えられた電力制約の中でどのセンサをどのタスクに割り当てるか、という問題を対象とする。

既存研究においては、観測タスクは「利得」「需要」というパラメータを持つものに抽象化され¹⁾、センサに割り当てられた際にセンサから「効用」を得る。効用がタスクの要求を満たした (例: 効用の和がタスクの需要を超えた) 際にシステムはタスクの利得を得る。ここで、需要は観測タスクがセンサからどのくらいの情報を必要とするかを抽象化したもので、効用は各々のセンサが取得したデータの、タスクに対する有用度を抽象化したものである。利得はタスクの優先度のように働き、利得の大きいタスクを小さいタスクより優先する

^{†1} 東京大学
The University of Tokyo

^{†2} 国立情報学研究所
National Institute of Informatics

ことが好ましい。このようにとらえると、問題は、与えられた電力資源制約下で得られる利得の和を最大化するセンサ・タスク間の割当てを決定することと定式化できる。

典型的 WSN では、データの収集箇所となるベースステーションが存在する。また、センサノードは搭載されたセンサを用いて観測を行うだけでなく、観測値を無線通信デバイスを用いてベースステーションに送信する必要がある。ここで、マルチホップ通信を行う際、送信元センサノードだけではなく、通信経路に存在しデータの中継するセンサノードも電力を消耗する。これらのノードが電力を消耗しきってしまうと、データの中継ができなくなってしまう。そのため、これらの通信に用いられる電力消費は無視できず、考慮する必要があると我々は主張する。

本論文では、まず上記の割当て問題を定式化する。この問題では、センシングに用いる電力だけでなく、通信に用いる電力も考慮される。次に、WSN に適した分散型の割当て決定アルゴリズムを提案する。このアルゴリズムではセンサノードとベースステーション間の通信路も考慮している。データの中継するノードのバッテリー切れを防ぐことによって WSN の生存時間を延長し、タスクの実行にともなって得られる利得の総和を増加させることができる。さらに、我々は目標となる WSN 稼働時間が与えられるケースにも取り組み、WSN 稼働時間内で利得を最大化するアルゴリズムを提案する。

我々は提案した手法をソフトウェアによるシミュレーションによって評価し、得られた総利得について既存手法との比較を行った。さらに問題を緩和して最適解を与え、提案手法の与える解と比較した。実験の結果、提案手法は既存手法と比較して、10%程度得られる総利得を向上させることができた。

本研究の貢献は以下のとおりである。(1) WSN のコンテキストで重要となる通信の要素に注目し、センサ・タスク間の割当て問題を再定義した。(2) 通信経路の残余電力に注目した分散割当てアルゴリズムを提案した。(3) 既存研究および理論的上限との比較実験を行い、既存手法より我々の手法が優れていることを示した。

本論文の残りの構成は以下のとおりである。次の章において、センサ・タスク割当てに関連する研究をまとめた。続く 3 章において、我々の対象とするセンサ・タスク間の割当て問題を説明する。4 章において、通信経路情報を加味した分散割当てアルゴリズムを提案する。5 章において、既存研究との比較実験を行い、提案手法が優位であることを示した。6 章において、提案手法の妥当性および結果について議論を行い、7 章において本研究をまとめた。

2. 関連研究

Rowaihy らによるセンサ割当てやセンサ選択に関するサーベイ¹⁴⁾では、問題を 4 つのカテゴリに分けて考えている。被覆問題・物体追跡問題・単一タスク割当て問題・複数タスク割当て問題である。本研究はこのうち、複数タスクへの割当てに取り組んだものとなっている。

単一タスク割当て問題および複数タスク割当て問題においては、タスクは単純化して扱われている。文献 4) においては、電力消費の問題を「効用」・「コスト」の枠組みでとらえ、2 つの要素がトレードオフの関係にあり、コストに関する制約がある中で効用を最大化する問題に取り組んでいる。また、文献 3) においては、タスクの定義とともに効用関数を与えられる枠組みを提案している。これらの研究では、WSN が 1 つのタスクを実行することを想定しており、複数タスクでの競合などは考えられていない。

いくつかの近年の研究において、複数タスクの WSN への割当て問題が考えられている^{1),7),16)}。文献 7) において、著者らは WSN 内に投入されるタスクは場所をパラメータとして持つものとし、最適化問題として定式化を行った。扱う問題に対する仮定として、すべてのタスクが 1 度に与えられるケース（静的）およびオンラインにタスクが出現するケース（動的）の双方を扱っている。静的な問題に対しては、近似アルゴリズムを反復して用い、徐々に解を改善するアルゴリズムを与え、貪欲法よりも優位であることを示している。動的な問題に対しては、残余電力の低下したセンサはより注意深く決定を行う分散手法を提案している。また、文献 16) では、著者らは、文献 7) における単純な効用関数を拡張しイベント検出および物体追跡問題に適した効用関数を提案した。同様に、文献 15) においては、文献 1) で提案された単純な制約式を拡張し、通信帯域に関する制約を考慮した問題に取り組んでいる。これらの研究では、いずれにおいても、通信にかかるコストは考慮されていない。これらの研究においては、バッテリーに関する制約がまったくないか、あるいはセンサノードのバッテリーはセンシングにのみ用いられるものと仮定しており、これは現実的な仮定とはいえない。我々の研究では通信に用いられる電力を考慮した、より現実的な電力資源モデルに取り組み、そのモデルに適した割当て手法を与える。

3. 問題設定

3.1 シナリオ例

図 1 に本論文で対象とするシステムの動作例を示す。システムは 1 つのベースステーショ

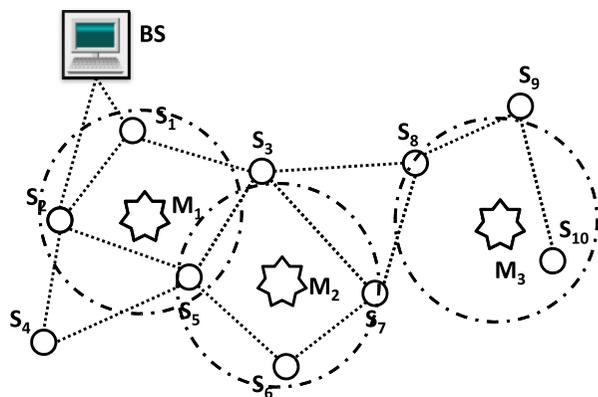


図 1 システム動作例

Fig. 1 Example scenario of the system.

ン (BS) と複数のセンサノード ($S_1 - S_{10}$) からなる．ここで、各センサノードは自ノードの位置を GPS や文献 5) などの手法によって取得することができるものとする．システムの利用者は観測タスク (M_1, M_2, M_3) を BS を通じてネットワークに投入する．観測タスクはカメラセンサでの観測といったものを想定しており、図中で星印によって示された観測対象となる位置を持つ．システムを敷設する時点で、対象となる環境の情報や主となる観測の目的 (例．生態観測、侵入検知) などは取得可能である．

ノードは通信可能半径を持ち、その外部のノードと通信できない (図においては点線で示された辺を通じてのみ通信ができる)．同様に各ノードは観測半径を持ち、その内部の対象を観測できる．図においては破線で示された円内のノードが中心にあるタスクを観測できる．各ノードは指向性を持つものとしており、ただだか 1 つの対象のみを観測できる．また、タスクのプリエンブションは対象とするシステムでは許されないものとする．すなわち、タスクに 1 度ノードを割り当てると、そのノードはタスク終了時まで観測する責務が生じるものとする．

タスクは観測期間を持ち、各々異なった時間にネットワークに投入される．そのため、システムはタスクを実行するノード集合を実行時において選択する必要がある．たとえば M_1 を実行するためには、 S_1, S_2, S_5 より実際に M_1 を観測するノードを選択する必要がある．同様にして M_2, M_3 にもノードを割り当てる必要があるが、いくつかの解決すべき問題が生じる．

まず、仮に S_5 を M_1 に割り当てた後に M_2 が投入されたとすると、 S_5 を M_2 に割り当てることができなくなってしまう． M_2 が M_1 よりも重要度が高い場合、これは好ましくない状態である．将来のタスク情報を知ることができないため、タスク開始時にはタスクのパラメータから注意深く割当てを決定する必要がある．また、 S_{10} を M_3 に割り当てた際、 S_{10} は観測したデータを BS に転送する必要がある．マルチホップで通信を行うため、 S_9 といった中継ノードの状態も M_3 の実行に関係する． S_9 のバッテリーが消耗してしまった際には、 S_{10} から BS への他の通信経路が存在しないため、 M_3 の実行ができなくなってしまう．この例は、割当て時に中継ノードの残余電力も考慮に入れる必要があることを示唆している．

次の 3.2 節において、このシステムに対する定式化を与える．

3.2 定式化

本論文で扱う問題を以下のように定義する．各センサノード $s_i \in S, (1 \leq i \leq n)$ はパラメータとして、二次元空間での設置箇所 (sx_i, sy_i) 、バッテリー残量 b_i を持つ．各タスク $m_j \in M, (1 \leq j \leq m)$ は同様に、観測位置 (mx_j, my_j) 、利得 p_j 、需要 d_j 、開始時刻 τ_j 、継続期間 λ_j 、要求データサイズ δ_j をパラメータとして持つものとする．タスクに割り当てられたセンサノードはタスクで指定された箇所を観測し、得られた情報をマルチホップ通信でベースステーションに送信する．この手続きを完了することでタスク m_j は、割り当てられたセンサノード s_i より、効用 e_{ij} を受け取る．センサノード s_i 、タスク m_j 間の効用 e_{ij} は、センサノード設置箇所 (sx_i, sy_i) と、タスク観測箇所 (mx_j, my_j) との間の距離から求まるものとする．これは、地理的な条件が情報の価値を定める主要因であるという仮定に基づいている．すなわち、より対象地点から近い場所で観測すると、より価値の高い情報が得られると仮定している．

システムがタスク m_j を実行して得られる利得 p_j に関しては様々な定義の仕方が考えられるが、我々は単純な関数定義 $p(y_j)$ を以下のように与えた．ここで y_j は効用合計 $\sum_i e_{ij}$ とタスクの要求 d_j の間の比である．

$$p(y_j) = \begin{cases} p_j, & \text{for } y_j \geq 1 \\ p_j \cdot y_j, & \text{for } y_j \geq T \\ 0, & \text{otherwise} \end{cases} \quad (1)$$

この関数の特徴として、2 つの点があげられる．まず、効用合計 $\sum_i e_{ij}$ が閾値 T (例． $T = 0.5$) を用いて、 $d_j \cdot T$ を超えない場合、システムは利得をまったく得ることができない．また、効用合計がタスク需要 d_j を超えた際、追加で得られる利得はない．1 つ目の特

徴は、閾値以下の不十分な情報は意味をなさないことを表し、2つ目の特徴は過剰な情報は観測精度を向上させないことを表している。

我々はセンサノードやタスクを上記のように定義し、さらに $x_{ijt} \in \{0, 1\}$ を時刻 t におけるセンサノード s_i のタスク m_j への割当て変数とする。また、 f_{ij}^k をセンサノード s_i から s_j のタスク m_k についての通信量とする。さらに、 σ_{jt} をタスク m_j の継続期間に関する指示変数とおくと、問題は式 (2) のように表現できる。

定式化した問題において、目的関数は利得 p の和であり、これを最大化したい。制約については、最初の制約式は y_{jt} と x_{ijt} の関係を示している。また、2番目・3番目の式は、各ノードおよびベースステーションにおけるデータ流量の満たすべき性質を表現している。4番目の式は電力制約を表現しており、5番目の式は1つのセンサノードはある時刻においてたかだか1つのタスクにのみ割り当てられることを表している。さらに6番目の式において、プリアンプションの禁止を条件として加えている。ここで、 $sense_i()$ 関数はセンサノード s_i におけるセンシングに用いた総電力量を返し、同様に、 $send_i()$ 関数、 $recv_i()$ 関数は送信および受信に用いた電力量を表す。各関数の定義は式 (8) から得られるものとする。また、 $(\mathbf{x}, \mathbf{f}, \Delta)$ は各々 x_{ijt} , f_{ij}^k , δ_j を要素として持つ集合である。

上記の問題を解くことによって問題に対する最適のセンサ割当てを求めることができる。しかし、現実にはタスクは動的に WSN に投入され、事前にパラメータを知ることができない。このため、上記の問題をそのまま解くのではなく別の方法によって割当てを求めることが必要となる。

$$\begin{aligned}
 \max : & \sum_t \sum_j \sigma_{jt} p(y_{jt}) \\
 s.t. : & \sum_i x_{ijt} e_{ij} \geq d_j y_{jt} \\
 & \sum_j x_{ikt} f_{ijt}^k - \sum_j x_{ikt} f_{jit}^k = x_{ikt} \delta_k \\
 & \sum_j f_{jit}^k = \sum_l x_{lkt} \delta_k \quad \text{for } i = \text{sink} \\
 & \sum_t sense_i(\mathbf{x}, \Delta) + \sum_t send_i(\mathbf{f}, \mathbf{x}, \Delta) + \sum_t recv_i(\mathbf{f}, \mathbf{x}, \Delta) \leq b_i \\
 & \sum_j x_{ijt} \leq 1
 \end{aligned} \tag{2}$$

$$\sum_t x_{ijt} = \lambda_j \quad \text{or } 0$$

$$\text{where : } x_{ijt} \in \{0, 1\}$$

$$y_{jt} \in [0, 1]$$

$$f_{ijt}^k \geq 0$$

$$\sigma_{jt} = \begin{cases} 1, & \text{for } \tau_j \leq t < \tau_j + \lambda_j \\ 0, & \text{otherwise.} \end{cases}$$

4. 経路情報を用いたセンサ割当て

本章において、我々は3章で提示した問題に対し割当てアルゴリズムを提案する。本手法では、通信経路における残余電力を考慮して割当てを行う。そのため、観測を行っているノードとベースステーション間の通信経路が長期間にわたって存在し、ノードが観測したデータをベースステーションに送ることが可能となる。長期間の運用によって、WSN はより多くのタスクを実行することができ、結果としてシステムは利得総和を向上させることができる。

4.1 アルゴリズムへの要求

まず、我々は問題に対する解法への要求を以下のように定義する。

分散型 ベースステーションが各ノードの情報を集め、すべての割当てを決定する集中型の割当て手法は、大量の通信を必要とするため WSN に適していない。このため、個々のセンサノードが周囲の情報から割当てを決定する分散型手法が問題設定に適している。通信路アウェア 文献 7) の問題設定とは異なり、我々は典型的 WSN シナリオを対象とし、タスクに割り当てられたノードは観測値をベースステーションに送信する必要がある。このことから、観測を行うセンサノードの残余電力とともに、観測値が送信される通信経路の残余電力についても同様に考慮に入れる解法であることが求められる。

4.2 アルゴリズム概略

本手法においては、各タスクの開始時刻において、各タスクの割当てを決定するリーダーノードを選出する。リーダーノードはセンサノードとタスクの地理的場所から決定される。各タスクは観測の対象となる地点情報をパラメータとして持つため、文献 8) にあるような手法を用いて最も観測地点に近いセンサノードを見つけることができる。

タスククエリを受け取ったリーダーノードは、タスク情報を周囲のノードにブロードキャストする。タスク情報はタスクに関する利得や需要といったパラメータを含んでいる。リーダー

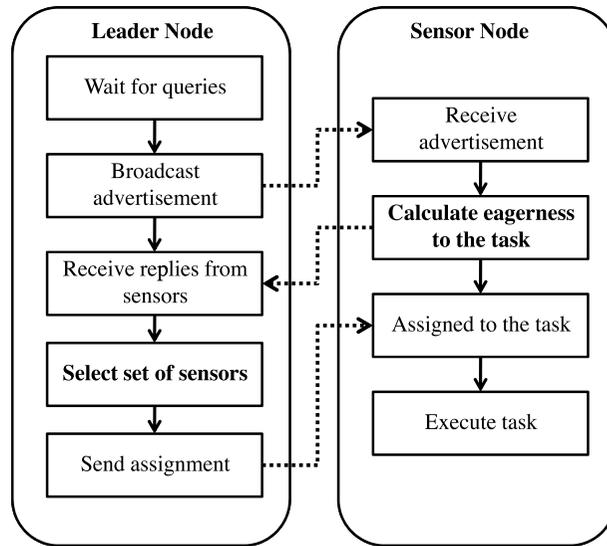


図 2 アルゴリズム概略図
Fig. 2 Algorithm flow diagram.

ノードがブロードキャストする最大のホップ数はセンサノードの通信範囲とセンシング範囲に基づいて変更することができる。タスク情報を受け取ったセンサノードはタスクへの参加傾向を、残余電力・タスク情報・経路情報（後述）から計算し、タスクへの寄与の期待値と比較することによってタスクへ参加するか否かを決定する。ここでの課題は、局所的情報からどのようにして適切な参加傾向の値を求めるかである。

タスクへの参加を決めたセンサノードはリーダノードに対しその旨を通知する。リーダノードはこれらの候補となるノードリストを各センサノードの効用について降順に並べ替え、ノードリストから部分集合を式 (1) に基づいて貪欲に選択し、実際に割当てノードとして決定する。本アルゴリズムの概略図を図 2 に示す。

4.3 経路情報の取得

我々は効率の良いセンサ・タスク割当てを決定するため経路情報が必要であることを主張した。割当て決定時に対象となる各ノードとベースステーションの間の経路情報をそれぞれ取得するためには経路長のホップ数に線形の通信量が発生する。このようにしてすべてのノードについて経路情報を求めると通信量を無視することができず、ノードの多くの電力を

消耗してしまう。我々はこの単純なアプローチとは異なり、対象とする WSN がクエリに応答するシステムであることを利用する。このシステムにおいて、WSN 利用者はタスクのクエリをベースステーションを経由してネットワークに投入し、クエリはベースステーションからマルチホップ通信を用いてリーダノードへと通知される。多くの WSN システム^{(12),(19)} は類似のシステムであるため、クエリがリーダノードへと送信されるという想定は現実的である。そのため、タスククエリがノードからノードへと経由されている際に経路情報を収集することができる。収集された経路情報はタスククエリに上乗せされ、リーダノードへと送られる。ここで、経路情報として我々は以下のデータを利用する。

- 経路中で最も少ない残余電力 $energy_p$
- 上ノードにおける現在の単位時間あたりの電力消費量 $load_p$

ここで、丸めを行うことでこれらの 2 つの値はそれぞれ 16 ビット整数値で表現できる。すると上乗せされるデータは 4 bytes であり、他のタスクパラメータ（位置情報、利得、需要、観測期間など）とあわせても、無線センサネットワークで標準的に用いられる TinyOS⁽¹⁰⁾ におけるパケットの既定データフィールド長 (= 28 bytes) 内に収まると考えられる。追加したデータはヘッダなどを含めた全体のパケットサイズと比較して少量であるため、追加コストは無視できる。

クエリを転送する中継ノードにおいては、受信したクエリに上乗せされている $energy_p$ と自ノードの残余電力を比較し、もし自ノードの残余電力が少ない場合には上乗せされた $energy_p$ とともに $load_p$ を更新し、次の中継ノードへと転送を行う。

留意すべきこととして、取得される経路情報はベースステーションとリーダノード間についての情報であり、各割当てセンサノードとベースステーション間の経路とは異なっている。しかし、あるノードとその近隣ノードにおけるベースステーションまでの経路の電力に関する性質は似通っていることが期待できるため、近似としてベースステーションとリーダノード間の経路情報を採用する。この手法の妥当性については 6 章で議論する。

4.4 各センサノードにおける決定

タスク情報をリーダノードから通知されたセンサノードは、そのタスクの候補ノードとなるか、またはタスクに参加しないかをノードの現在の状況とタスク情報から決定する。ノードの残余電力が減少してきた際は、低い利得しかもたらさないタスクは無視し、より高い利得のタスクを優先する方が好ましい。この性質を表現するため、タスクへの参加傾向を定義し、その値を限られた情報から適切に操作することを提案する。

我々は各ノードにおける残余電力より、経路における残余電力に重みをつけて参加傾向 w

を計算する．式 (3) において, $energy_p$, $energy_s$ および B は各々, 経路における残余電力, ノードにおける残余電力, システム運用を開始した時点での各ノードの電力とする．なお簡単のため各ノードの開始時点における電力を同一であるものとした．もともとのノードへの供給電力が同一である場合, この仮定は妥当である．

$$w = q \cdot \left(\frac{energy_p}{B} \right)^r + (1 - q) \cdot \frac{energy_s}{B} \quad (3)$$

$$\frac{u}{d} \cdot p \cdot w \quad (4)$$

各センサノードはタスクを実行した際の効用を計算できる．この効用から, センサノードは式 (4) のように, タスクへの期待される寄与と参加傾向の積を計算することができる．

式 (4) で求めた値はセンサノードの寄与の期待値 (式 (5)) と比較される．式 (4) での値が期待値を上回った場合, センサノードはこのタスクの候補ノードとなり, リーダノードに通知を行う．ここで我々は, タスクパラメータの期待値は取得可能だとしている．3.1 節で述べたように, システムが敷設環境や実行するタスクの大まかな性質を事前に知っているとき, これは自然な前提である．

タスク情報を受け取ったリーダノードの周囲のノードがリーダノードに通知を完了した後, リーダノードは実際に割当てノードを決定し, タスクを実行する．

$$\frac{E[u]}{E[d]} \cdot E[p] \quad (5)$$

4.5 目標稼働時間が与えられた際の割当て手法

前述のアルゴリズムは電力制約が与えられた際の利得総和を最大化することを目的としていた．この節では, WSN の稼働時間が与えられたケースに取り組む．すなわち, このケースでは目標とする WSN の稼働時間が事前に決まっており, その期間に対してはある程度余剰の電力が与えられているものとする．その際, システムの目的は, 与えられた稼働時間内で得られる利得総和を最大化することとなる．

式 (3) において, タスクへの参加傾向は残余電力に従い減少する．このことによって, ネットワークの連結性が保たれ, ノードは将来の, より利得の高いタスクを実行することができる．しかし, 本節のケースにおいては, 稼働時間が限られているため, 残りの稼働時間を考慮することでより良い割当てを与えることができる．

多くの余剰な電力資源がある場合, 稼働時間内により多くの電力を消費した方が良いため, 残りの稼働時間が短くなってきたときには, センサノードは低い利得のタスクにも積極的

に参加した方が好ましい．ここで, 現在の経路における電力消費から, 有効稼働時間 t_{avail} を式 (6) に基づいて求めることができる．これらの項を利用して, 参加傾向 w_{life} を新しく式 (7) のように定義する．ここで, $load_s$ は対象としているセンサノード上での単位時間あたりの電力消費量, t_{rem} は残りの WSN 稼働時間を示す．求めた w_{life} は前節の w と同様に用いられ, 期待値との比較によってセンサノードは候補ノードとなるかどうかを決定する．

$$t_{avail} = \min \left(\frac{energy_s}{load_s}, \frac{energy_p}{load_p} \right) \quad (6)$$

$$w_{life} = w \cdot \left(1 + \left(\frac{t_{avail}}{t_{rem}} \right)^v \right) \quad (7)$$

5. 評価

5.1 シミュレーション設定

我々は, 以下の設定で提案手法を評価した．ネットワークに関するパラメータ (表 1 に記載) は文献 7) と共通である．クエリおよび観測値のルーティングの方法としては最短経路木を用いた．

センサノード・タスクは環境にランダムに配置されるものとし, ベースステーションは二次元環境の左上隅に配置されているものとする．タスクの発生はポアソン分布に従い, 1 時間に平均で 16 個のタスクが発生するものとした．タスク継続期間 λ_j , タスク需要 d_j およびタスク利得 p_j は指数分布に従うものとし, 平均値は各々 1 時間, 2.0, 10.0 とした．さらに, 各パラメータに上限と下限を設定した．タスク継続時間は 5 分から 4 時間の値をとるものとし, 需要の下限は 0.5, 利得は 0 から 100 の値をとるものとした．また, 要求データサイズ δ_j は各タスクで共通のものとして, 100 bits であると聞いた．

電力モデルは文献 6) のものを採用した． $E_{sense}(b)$ は bbit のセンシングに用いる電力を

表 1 シミュレーション設定
Table 1 Simulation settings.

Field size	400 × 400 (m)
Number of nodes	500
Communication range	30 (m)
Sensing range	40 (m)
Routing	Shortest path tree

表 2 電力モデルに用いたパラメータ設定
Table 2 Parameter settings for electricity.

V_{sup}	Supply voltage to sensor	2.7 (V)
p	Packet size	2 (kbit)
E_{elec}	Energy dissipation : electronics	50 (nJ/bit)
E_{amp}	Energy dissipation : power amplifier	100 (pJ/bit/m ²)
T_{sense}	Time duration for sensing	0.5 (mS)
I_{sense}	Current for sensing activity	25 (mA)
B	Battery capacity of one node	750 (mAh) · 3600 (S) · 1.5 (V)

表し、また、 $E_{tx}(p, dist(i, j))$ および $E_{rx}(p)$ は、 p bit のパケットを s_i, s_j 間で送信・受信する際に (s_i が) 消費する電力を表している。ここで、 $dist(i, j)$ は、 s_i, s_j 間のユークリッド距離を示している。式 (8) は電力消費の定義式であり、表 2 は電力消費を計算するための定数を掲示している。ここで、 $k = 2$ とした。

$$\begin{aligned} E_{sense}(b) &= bV_{sup}I_{sense}T_{sense} \\ E_{tx}(p, dist(i, j)) &= pE_{elec} + p \cdot \{dist(i, j)\}^k \cdot E_{amp} \\ E_{rx}(p) &= pE_{elec} \end{aligned} \quad (8)$$

式 (9) はセンサノードが寄与する効用を定義している。効用はセンサノード s_i とタスク t_j との距離 D_{ij} に基づいている。ここで、定数 c は 60 とし、 r_{sense} は最大センシング半径を表している。

$$e_{ij} = \begin{cases} \frac{1}{1+D_{ij}^2/c} & (\text{if } D_{ij} \leq r_{sense}) \\ 0 & (\text{otherwise}) \end{cases} \quad (9)$$

我々は目標となる WSN 稼働時間が与えられないケース、与えられるケースの 2 つのケースを扱った。各々のケースについて、4 章で提案した手法を用いた。ここで、式 (3) および式 (7) について、 $q = 0.6$, $r = 2$, $v = 2$ とした。評価は、得られた利得総和および生存ノードの割合に基づいて行った。

比較として下記の手法を採用した。

貪欲法 (greedy) タスク情報をリーダーノードから受け取った未割当てなノードはすべて候補ノードとなる。リーダーノードはセンサノードの寄与を降順に並べ、割当てを決定する。

既存手法 (previous)⁷⁾ センサノードはタスクの候補となるか否かを現在の残余電力から決定する。残余電力が減少するに従い、センサノードはより高い利得を持つタスクを優

先し、低い利得を持つタスクを無視するようになる。また、この手法はタスクの継続時間の期待値を知っているという仮定の下、目標となる稼働時間が与えられた際には、その値を利用する。

緩和問題の解 (relaxed) 式 (2) で提示した問題は各時間に付随した変数があり、変数の数が膨大となるため、そのまま解くのが困難である。そこで、我々はすべてのタスクが同時に発生したものと仮定して変数の数を減らし、さらに式 (2) に存在した整数制約を緩和して通常の線形計画問題 (式 (10)) とした。この問題を解くことで、もとの式 (2) で与えられた問題の解の上限を与えることができる。なお、我々はこの緩和問題を解くために lp_solve²⁾ を用いた。

$$\begin{aligned} \max : & \sum_j p_j y_j \lambda_j \\ \text{s.t.} : & \sum_i x_{ij} e_{ij} \geq d_j y_j \\ & \sum_{j,k} x_{ik} f_{ij} - \sum_{j,k} x_{ik} f_{ji} = \sum_k x_{ik} \delta_k \lambda_k \\ & \sum_j f_{ji} = \sum_{l,k} x_{lk} \delta_k \lambda_k \quad \text{for } i = \text{sink} \\ & \text{sense}_i(\mathbf{x}, \Delta) + \text{send}_i(\mathbf{f}, \mathbf{x}, \Delta) + \text{recv}_i(\mathbf{f}, \mathbf{x}, \Delta) \leq b_i \\ \text{where} : & x_{ij} \in [0, 1] \\ & y_j \in [0, 1] \\ & f_{ij} \geq 0 \end{aligned} \quad (10)$$

5.2 目標稼働時間がない場合の結果

前節で提示した手法と提案手法を比較するため、ソフトウェアによりシミュレーションを行った。図 3、図 4 および表 3 は各々、各タイムステップでの達成利得の割合、各タイムステップにおける生存ノード数の割合および獲得した利得総和を各々提示している。ここで達成利得の割合は、タスクを実行して得ることができた利得の総和と、そのタイムステップにおいて存在するタスクの利得総和の比である。

図 3 によると、貪欲法は最も性能が低く、達成利得の割合もすぐに落ち込んでしまう。400,000 ステップ後における貪欲法の達成利得割合は 20% 程度である。既存手法は貪欲法に比べ、センサノードの残余電力を割当てに加味しているため良い結果を出しており、生存時間も長く性能の落ち込みは穏やかである。提案手法はさらに高い結果を示し、300,000 ス

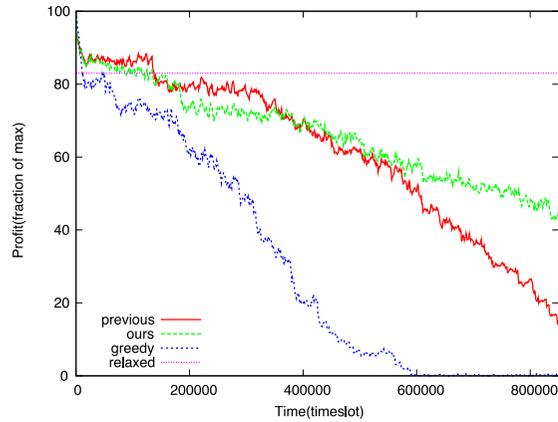


図 3 各時刻における達成利得の割合
Fig. 3 Fraction of achieved profit.

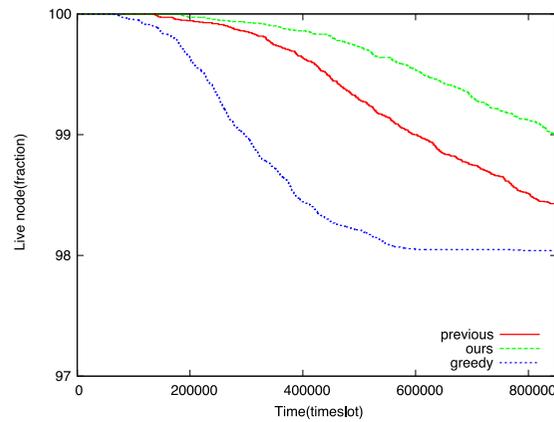


図 4 各時刻における生存ノードの割合
Fig. 4 Fraction of live nodes.

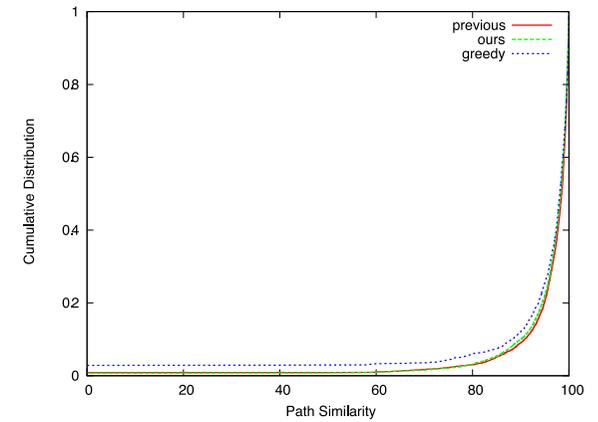


図 5 残余電力に基づく類似度の累積分布
Fig. 5 Cumulative distribution of similarity based on residual energy.

表 3 目標稼働時間がない場合における利得総和の比較
Table 3 Comparison of profit gained without target lifetime.

Method	Overall Profit
Greedy Method	6.88×10^7 (1)
Previous Method	1.44×10^8 (2.09)
Proposed Method	1.59×10^8 (2.31)
Solution for Relaxed Problem	2.03×10^8 (2.96)

表 4 目標稼働時間が与えられた場合の利得総和の比較
Table 4 Comparison of profit gained with target lifetime.

Method	400,000 TS	600,000 TS
Previous Method	8.25×10^7 (1)	9.46×10^7 (1)
Proposed Method	8.48×10^7 (1.03)	1.07×10^8 (1.13)

テップあたりでの達成利得割合は既存手法に負けているものの、性能の落ち込みはさらに緩やかで、特に実行の後半において差が顕著になっている。この性能の伸びは、通信路における残余電力を考慮したためである。図 3 では緩和問題の最適解における平均の達成利得割合も掲示している。提案手法での利得総和は最適解で得られる利得総和の 78% である。図 4 に示した生存ノード数の割合でも同様の傾向が見られる。貪欲法はセンサノードの資源を早い段階で消耗してしまっているが、既存手法・提案手法ではこの順に、より長くセンサノードの寿命を保っていることが分かる。

表 3 に示した利得総和から、既存手法は貪欲法に比べ 2 倍程度の利得が得られていることが分かる。これは、ノードにおける残余電力を考慮したことによるものである。さらに、

提案手法は 10.5% ほど既存手法より得られる利得が大きい。これは通信路における残余電力を考慮した効果である。

5.3 目標稼働時間が与えられた場合の結果

目標稼働時間として 400,000 および 600,000 ステップが与えられた際の総利得について、ほかのパラメータは変えずに実験を行った。得られた結果は表 4 に掲示した。

目標となる稼働時間が与えられない際、図 3 から分かるように、400,000 ステップにおける提案手法の利得総和は既存手法のそれよりも低いものとなっている。対照的に、稼働時間が与えられた際には提案手法は既存手法より高い利得を獲得できている。これは、提案手法においては、残余稼働時間が短くなってきたときにタスクへの参加傾向が上昇するためである。稼働時間に関する情報がない場合には、センサノードは残余電力が減るにつれ参加傾向は減少するが、稼働時間が定まっている際には、ある時点において余剰電力が残り時間に対

し過剰となるため、より積極的にタスクに参加することが好ましい。この評価で得られた結果により、4.5 節で提案したアルゴリズムの変更が有効であることがいえる。

6. 議 論

本章では、提案手法で用いた経路情報の妥当性および対象問題の解の上限について議論を行う。

6.1 経路情報の妥当性

提案手法において、我々は割当て対象となるノードからベースステーションまでの経路をベースステーションからリーダノードまでの経路で近似した。ここでの近似が正しく行えず、残余電力があまり残っていないと判断される場合においては、タスクへの参加傾向が減少し、小さい利得を持ったタスクが実行されないことが考えられる。このケースにおいては本来得られる利得を失ってしまう。逆のケースはより悪く、残余電力の少ない経路に負荷を与えすぎて、ネットワークが途切れてしまうことが考えられる。図 5 は元の経路と近似経路の類似度を示したものである。各割当て手法を用いた際に、割当て対象となるノードとベースステーション間およびリーダノードとベースステーション間の通信路の類似度を図示した。ここで、類似度は各通信路における平均の残余電力をもとに評価した。すべてのケースにおいて、残余電力に関して高い類似度があり、近似した経路の残余電力が正しい残余電力と大きく異なるケースが少ないことが分かる。

我々は、最短経路木がルーティングとして用いられていることを仮定した。文献 18) で提示されたルーティングにおいては、ネットワークの連結性を保つため、ノード間での電力消費を均等化することが目指されている。電力消費がよく均等化されたケースにおいては、近似経路と実際の経路での残余電力の差がほとんどないことが予想されるため、提案手法がうまく働くことが予想される。

6.2 解の上限

対象とした WSN において、ベースステーション周囲のノードがボトルネックとなることが考えられる。これらのノードが電力を消耗しきった際、ネットワーク内の残りのノードがベースステーションにアクセスできなくなるためである。このことから、5 章での問題設定における解の平均的上限を与えることができる。式 (11) はこれらのノードの持つ平均的電力 $b_{bottleneck}$ を与える。ここで、 r_{comm} は通信半径を示し、 ρ はノード密度を表す。さらに、式 (12) は、1 パケットの送受信にかかる電力 c_{unit} を表している。よってベースステーション付近のノードが活動できる最大の期間は 2 つの項の比で表される。

$$b_{bottleneck} = \frac{1}{4} \cdot \pi \cdot r_{comm}^2 \cdot \rho \cdot B \quad (11)$$

$$c_{unit} = E_{tx}(p, E[dist]) + E_{rx}(p) \quad (12)$$

前章で述べたとおり、タスクの需要の下限は 0.5 であり、センサノードは最大で 1 の効用をタスクに与えることができる。理想的には 1 ノードでタスクの需要を満たすことができる。また、利得の最大値は 100 であることを用い、 $b_{bottleneck} = 1.59 \times 10^4 [J]$ および $c_{unit} = 3.39 \times 10^{-4} [J]$ を得る。そのため、解の平均的上限は、 4.69×10^9 となる。これは 5 章で得られた結果よりかなり大きい。

この利得総和を得るためには、システムは非常に小さい需要かつ高い利得を持ったタスクを待つ必要がある。付近に利用可能なセンサが存在した際、システムはこのタスクを実行し、そうでなければ無視するという動作をすることでここで与える利得が達成できる。この際、多くのタスクが無視され、実行されない。このように、より現実的な運用を想定すると利得総和の最大化のみでは十分でないことが考えられる。4 章で扱ったように限られた時間で利得総和を最大化したり、あるいは実行しなかったタスクの利得をペナルティ項として扱ったりするなど、各目的にあわせ目的関数を調整することは有用であると考えられる。

7. ま と め

WSN においては、センサノードで得られた情報はアプリケーションでの利用などのために、ベースステーションに送信される必要がある。このためにはセンサノードは無視できない通信コストを消費する必要がある。したがって、センシングに用いるセンサノードを保全するだけでは不十分であり、センシングを行っているノードとベースステーション間の通信経路上の経由ノードについても考慮する必要がある。本研究ではこの問題を 3 章のように定式化した。この定式化においては、資源制約が単純なモデル⁷⁾とは異なって、通信に用いる電力資源も関わる。我々はこの問題に対し、通信経路の情報を近似し、タスククエリに上乘せし、割当て決定に利用する手法を提案した。ベースステーションが各センサ情報を収集し割当てを決定するアプローチとは異なり、提案手法はより現実の WSN に適した分散手法である。

シミュレーションによる結果により、得られる総利得について、提案手法が既存手法に比べ 10%程度性能が良いことが示された。さらに、提案アルゴリズムに小さな変更を加えることによって、目標となる WSN 稼働時間が定まっているケースにも対応できる。このケー

スでは、既存手法に比べて 3–13%程度の性能向上を達成することができた。

今後の研究課題としては、様々なルーティング設定のもとでの本手法の性能評価、および 6.2 節で論じたように、より現実的設定における目的関数のもとで問題をとらえなおすことがあげられる。

参 考 文 献

- 1) Bar-Noy, A., Brown, T., Johnson, M.P., Porta, T.F.L., Liu, O. and Rowaihy, H.: Assigning Sensors to Missions with Demands, *ALGOSENSORS*, Kutylowski, M., Cichon, J. and Kubiak, P. (Eds.), Lecture Notes in Computer Science, Vol.4837, pp.114–125, Springer (2007).
- 2) Berkelaar, M., Eikland, K., Notebaert, P., et al.: lpsolve: Open source (mixed-integer) linear programming system.
- 3) Bian, F., Kempe, D. and Govindan, R.: Utility based sensor selection, *IPSN*, Stankovic, J.A., Gibbons, P.B., Wicker, S.B. and Paradiso, J.A. (Eds.), pp.11–18, ACM (2006).
- 4) Byers, J.W. and Nasser, G.: Utility-based decision-making in wireless sensor networks, *MobiHoc*, pp.143–144, ACM (2000).
- 5) Goldenberg, D.K., Bihler, P., Yang, Y.R., Cao, M., Fang, J., Morse, A.S. and Anderson, B.D.O.: Localization in sparse networks using sweeps, *MOBICOM*, Gerla, M., Petrioli, C. and Ramjee, R. (Eds.), pp.110–121, ACM (2006).
- 6) Halgamuge, M., Zukerman, M., Ramamohanarao, K. and Vu, H.: An estimation of sensor energy consumption, *Progress In Electromagnetics Research B*, Vol.12, pp.259–295 (2009).
- 7) Johnson, M.P., Rowaihy, H., Pizzocaro, D., Bar-Noy, A., Chalmers, S., Porta, T.F.L. and Preece, A.D.: Frugal Sensor Assignment, *DCOSS*, Nikolettseas, S.E., Chlebus, B.S., Johnson, D.B. and Krishnamachari, B. (Eds.), Lecture Notes in Computer Science, Vol.5067, pp.219–236, Springer (2008).
- 8) Karp, B. and Kung, H.T.: GPSR: Greedy perimeter stateless routing for wireless networks, *MOBICOM*, pp.243–254 (2000).
- 9) Kulkarni, P., Ganesan, D. and Shenoy, P.J.: The case for multi-tier camera sensor networks, *NOSSDAV*, chi Feng, W. and Mayer-Patel, K. (Eds.), pp.141–146, ACM (2005).
- 10) Levis, P., Madden, S., Polastre, J., Szewczyk, R., Whitehouse, K., Woo, A., Gay, D., Hill, J., Welsh, M., Brewer, E. and Culler, D.: TinyOS: An Operating System for Sensor Networks, *Ambient Intelligence*, pp.115–148, Springer Berlin Heidelberg (2005).
- 11) Madden, S., Franklin, M.J., Hellerstein, J.M. and Hong, W.: The Design of an Acquisitional Query Processor For Sensor Networks, *SIGMOD Conference*, Halevy, A.Y., Ives, Z.G. and Doan, A. (Eds.), pp.491–502, ACM (2003).
- 12) Madden, S., Franklin, M.J., Hellerstein, J.M. and Hong, W.: TinyDB: An acquisitional query processing system for sensor networks, *ACM Trans. Database Syst.*, Vol.30, No.1, pp.122–173 (2005).
- 13) Ni, L.M., Liu, Y., Lau, Y.C. and Patil, A.P.: LANDMARC: Indoor Location Sensing Using Active RFID, *Wireless Networks*, Vol.10, No.6, pp.701–710 (2004).
- 14) Rowaihy, H., Eswaran, S., Johnson, M., Verma, D., Bar-Noy, A., Brown, T. and LaPorta, T.: A survey of sensor selection schemes in wireless sensor networks, *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, Vol.6562, p.35 (2007).
- 15) Rowaihy, H., Johnson, M., Eswaran, S., Pizzocaro, D., Bar-Noy, A., LaPorta, T., Misra, A. and Preece, A.: Utility-Based Joint Sensor Selection and Congestion Control for Task-Oriented WSNs, *Proc. Asilomar Conference on Signals, Systems & Computers* (2008).
- 16) Rowaihy, H., Johnson, M.P., Pizzocaro, D., Bar-Noy, A., Kaplan, L.M., Porta, T.F.L. and Preece, A.D.: Detection and Localization Sensor Assignment with Exact and Fuzzy Locations, *DCOSS*, Krishnamachari, B., Suri, S., Heinzelman, W.R. and Mitra, U. (Eds.), Lecture Notes in Computer Science, Vol.5516, pp.28–43, Springer (2009).
- 17) Satyanarayanan, M.: Pervasive Computing: Vision and Challenges, *IEEE Personal Communications*, Vol.8, No.4, pp.10–17 (2001).
- 18) Schurgers, C. and Srivastava, M.: Energy efficient routing in wireless sensor networks, *IEEE Military Communications Conference (MILCOM 2001)*, *Communications for Network-Centric Operations: Creating the Information Force*, Vol.1 (2001).
- 19) Yao, Y. and Gehrke, J.: The Cougar Approach to In-Network Query Processing in Sensor Networks, *SIGMOD Record*, Vol.31, No.3, pp.9–18 (2002).
- 20) Yu, L., Wang, N. and Meng, X.: Real-time forest fire detection with wireless sensor networks, *Proc. International Conference on Wireless Communications, Networking and Mobile Computing 2005*, pp.1214–1217 (2005).

(平成 22 年 5 月 21 日受付)

(平成 22 年 11 月 5 日採録)



鳥海 晋

2008年東京大学理学部情報科学科卒業，2010年東京大学大学院情報理工学系研究科コンピュータ科学専攻修士課程修了，同年同博士課程進学，国立情報学研究所アーキテクチャ科学研究系リサーチアシスタント，現在に至る．無線センサネットワークの研究に従事．



本位田真一（フェロー）

1978年早稲田大学大学院理工学研究科修士課程修了．(株)東芝を経て2000年より国立情報学研究所教授，2004年より同研究所アーキテクチャ科学研究系研究主幹を併任，現在に至る．2008年より同研究所先端ソフトウェア工学・国際研究センター長を併任，現在に至る．2001年より東京大学大学院情報理工学系研究科教授を兼任，現在に至る．現在，早稲田大学客員教授，英国UCL客員教授を兼任．2005年度パリ第6大学招聘教授．工学博士（早稲田大学）．1986年度情報処理学会論文賞受賞．日本ソフトウェア科学会理事，情報処理学会理事を歴任．ACM日本支部会計幹事，情報処理学会フェロー，日本ソフトウェア科学会編集委員長，日本学術会議連携会員．