

水中ワイヤレスセンサネットワークにおける伝搬遅延を考慮したタイムスロットスケジューリング

浜田 龍平¹ 村田 佳洋^{1,a)}

受付日 2012年11月8日, 再受付日 2012年12月27日/2013年2月4日/2013年3月5日,
採録日 2013年4月14日

概要: 近年, 海洋資源調査や災害感知のために, 水中用のワイヤレスセンサネットワーク (Underwater Wireless Sensor Network, 以下 UWSN) が研究されている. 水中では電波通信が効果的でないため, 音響通信がよく用いられる. 音響通信は伝搬遅延が大きく, センサノードごとの受信タイミングが異なるため, 水中環境のためのパケット衝突制御手法が必要である. 本論文では, UWSN における伝搬遅延を考慮したタイムスロットスケジューリング手法を提案する. 本手法は, 遺伝的アルゴリズムによって通信優先度を割り当て, タイムスロットを割り当てる. 提案手法を評価するためシミュレーション実験を行った. その結果, データ分割法と比較して, 58%少ないタイムスロット数での割当てを得ることができた.

キーワード: 水中ワイヤレスセンサネットワーク, 遺伝的アルゴリズム, タイムスロット

Considering the Propagation Delay Time Slot Scheduling for Underwater Wireless Sensor Networks

RYUHEI HAMADA¹ YOSHIHIRO MURATA^{1,a)}

Received: November 8, 2012, Revised: December 27, 2012/February 4, 2013/March 5, 2013,
Accepted: April 14, 2013

Abstract: Recently, Underwater Wireless Sensor Network (hereafter, UWSN) is studied. In the water, radio communication method is not efficient, then acoustic communication is usually used. In order to overcome long propagation delay, collision prevention control methods are needed. In this paper, we propose a time-slot scheduling method for UWSN. In this method, the communication priorities of nodes are determined using genetic algorithm, and the time-slots are allocated. To evaluate our method, we compared it with the data divided method in simulation. As a result, our method achieved about 58% less time-slots than that of comparative method.

Keywords: underwater wireless sensor network, genetic algorithm, time slot

1. はじめに

近年, 農業管理や工業システム, 災害感知のための環境観測による情報収集アプリケーションを実現する 1 つの手段として, ワイヤレスセンサネットワーク (Wireless Sensor Network, 以下 WSN) が注目されている. WSN では, 観

測したい環境に安価な多数のセンサノードと呼ばれるものを配置し, それらを実無線マルチホップ通信しノードがセンシングによって得た情報を交換することで大規模な環境観測を実現する研究が行われている.

WSN では, 一般的に電波による通信が用いられる. 電波通信における問題の 1 つとして, 伝搬遅延の問題がある. この伝搬遅延の問題を解決するため, PSFQ (Pump Slowly and Fetch Quickly) によるホップバイホップ通信を用いた手法がある [1]. これは, TCP がタイムアウトしてしまいデータを確実に伝搬できないという問題を効率

¹ 広島市立大学大学院情報科学研究科システム工学専攻
Department of Systems Engineering, Graduate School of Information Sciences, Hiroshima City University, Hiroshima 731-3194, Japan

^{a)} yosihim@hiroshima-cu.ac.jp

的な符号化を用いることで可能とした手法である。他に、CODA (Congestion Detection and Avoidance) を用いた手法がある [2]。これは、パケット衝突によるデータの損失を防ぐために CODA における開ループと閉ループの動作を組み合わせることによって高速な動作と効率的な状態を維持しデータの損失を防ぐ手法である。

また、WSN では効率的な通信を行うため、時間分割多元接続 (TDMA: Time Division Multiple Access), 周波数分割多元接続 (FDMA: Frequency Division Multiple Access), 符号分割多元接続 (CDMA: Code Division Multiple Access) などさまざまな MAC プロトコルが提案されている。しかし、水中では帯域幅が制限されているため、周波数分割多元接続 (FDMA) および符号分割多元接続 (CDMA) は効率的な通信ができない。このため、UWSN では帯域幅に依存しない時間分割多元接続 (TDMA) を利用した研究がさかんである。また、時間分割多元接続 (TDMA) では、伝搬遅延が問題とされる。

現在、WSN では、水中に応用させた UWSN (Underwater Wireless Sensor Network) がある。UWSN のアプリケーション例としては、海洋データの収集、海洋資源調査、汚染調査、海洋生態調査、軍事目的のものがある。これらの海底調査を行うことで、汚染拡散の抑制、火山の爆発の予測、地震の震源の特定などが行える。WSN では、一般的に電波による通信を行っているが、水中では電波の減衰が著しい。また、利用できる通信帯域幅が限られてくるため、UWSN では音響による通信が一般的である。

音響による通信では、電波による通信と同様に、干渉や伝搬遅延の問題があり、さらには、ドップラー効果が問題とされている。水中における音響通信では、伝搬遅延が著しく、センサノードの受信タイミングが異なる。また伝搬遅延の問題は、通信におけるパケット衝突に対しても非常に深刻な問題であり、パケット衝突制御は非常に困難な問題である。

そこで我々は、遺伝的アルゴリズム (GA: Genetic Algorithm) を用い通信における優先度を決定し、データ収集経路と優先順位から伝搬遅延を考慮した水中ワイヤレスセンサネットワーク (UWSN) のための効率的なタイムスロットスケジューリングを提案する。

本研究では、GA によって通信における優先度を割り当て、データ収集経路の考慮による優先度の更新とパケット衝突の可能性を考慮したタイムスロットの割当てを 3 段階に分けているところに特色がある。また、提案手法を評価するためシミュレーション実験を行った。その結果、データ分割法と比較して、58%程度少ないタイムスロット数での割当てを得ることができた。

2. 関連研究

WSN の前身には、Pister らによって開発、発表された

SmartDust [3] の存在がある。SmartDust は 1~2 ミリ四方のノードを空中にばら撒き、各々のノードが相互通信することによって、環境情報などを収集するといったコンセプトを持つ。この情報収集の形が現在の WSN の基本型となるものとされている。

現在、WSN の利用には軍事、環境観測などさまざまなアプリケーション先が考えられ、期待されている。たとえば、農業用地において、温度センサや湿度センサ、照度センサなどを備えたセンサノードを配置し、データ収集することで、農業用地の管理や生産を支援することが可能となる [4]。また、工業の監視システムにも利用されている [5]。これは、無線監視システムを多数配置することで、産業プロセスに関する情報を集めるものである。大林は、WSN の現状についてその周辺のハードウェアやソフトウェアに関して調査し、その具体的なアプリケーションを提案している [6]。

具体的なアプリケーションとして、斜面の挙動監視、落石検知、道路構造物の損傷検知などをあげている。関連して、歴史的建造物を保存するためにその背後の斜面などに実験的に WSN を設置し、その斜面の崩壊を予測する提案をしている [7]。これは、土壌の水分挙動をセンシングすることで斜面の状態を観測するというものである。

現在、WSN は水中に応用され、UWSN として研究がされている。UWSN のアプリケーション例としては、海洋環境観測、海底資源調査などが考えられる。これらの観測により、石油や石炭に代わる新たなエネルギーの発見やレアメタルなどの発見が期待される。しかし、水中では、帯域幅が制限されることや海流の影響、センサノードの汚染・腐食、リアルタイム性などさまざまな問題があげられる [8]。

Akyildiz らは、UWSN におけるさまざまなアプリケーションに対して、効率的なデータ通信のために UWSN のアーキテクチャを提案している [9]。この文献では、海面付近に存在するセンサノードの海流によるセンサノードの移動を問題視している。

Lu らは、UWSN における時刻同期手法を提案している [10]。この文献では、UA-TSP と呼ばれる時刻同期手法により伝搬遅延を保証し、シミュレーション実験により通信効率と消費電力がより最適となるような時刻同期を達成している。

また、Liu は、センサノードにあまり移動が見られないようであるならば、頻繁な時刻同期が必要でないとし、データ経路生成時間を短縮している [11]。

Che らは、浅い海域での電波通信を用いた時間分割多元接続 (TDMA) を利用した手法を提案している [12]。この文献では、小規模なネットワークにおいて VLF 技術を利用し、TDMA におけるタイムスロットスケジューリングを提案し、さまざまなネットワークシナリオに対する性能

を評価している。

提案手法のアプリケーションでは、一定の場所の水温などをつねにセンシングするような UWSN を想定する。提案手法では、データ収集のスケジュールを GA によって出力し、パケット衝突を回避しつつ通信伝搬遅延による空き時間を利用して通信時間を割り当てる。割り当ての際に、通信の優先度およびデータ収集経路をそれぞれ GA の染色体とし、効率良く求める。

提案手法は、複数の場所をセンシングしてデータを収集するアプリケーションを対象とし、通信の優先度とデータ収集経路を同時に GA を用いて調整することにより、より良い組合せを見つける点に特色がある。

3. 問題定義

3.1 概要

本研究では、以下のような海底における環境調査のための UWSN アプリケーションを想定する。図 1 のようにフィールド上に基地局ノード s_0 を含んだセンサノードが配置され、各センサノードは周辺の温度などの環境データをセンシングによって周期的に計測する。計測された環境データは、マルチホップ通信によって基地局ノード s_0 へとデータを集約される。その際、ノード s_5 とノード s_4 が同時にデータを送信すると、ノード s_2 で受信する際にパケットが衝突してしまう。本研究における問題の目的は、すべてのセンサノードから基地局ノードへデータを収集するまでの時間を最小化することである。なお海底では、海流の影響も少ないとして、本研究において設置されたセンサノードが海流によって移動することは考慮しない。また、海底は水平なものとし、海嶺や海溝などの音響通信を妨げる障害物がないものとする。

3.2 仮定

UWSN におけるセンサはすべてネットワーク上のノードであると見なし、ネットワークはセンサノードと基地局ノードの 2 種類によって構成される。センサノードは、センシングを行うことで周囲の情報を収集することが可能であり、データの集約をしながらマルチホップ通信によって基地局ノードへセンシングデータを送信する。このとき、他のセンサノードが自身のセンシングデータとともに

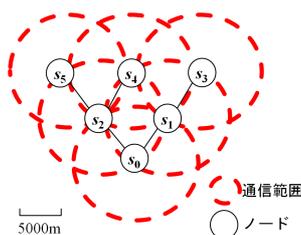


図 1 ノードの配置図
Fig. 1 Positions of nodes.

に中継して基地局ノードへとセンシングデータを収集する。基地局ノードは、全センサノードからセンシングデータを収集しており、基地局ノードからなるデータ収集経路の木をデータ収集木とする。なお、データを収集する際は、データ収集木上における子ノードからデータが送信されていないとき、親ノードはデータを送信しないとする。ここで、UWSN におけるすべてのセンサノードの集合を、 $S = \{s_0, s_1, s_2, \dots, s_n\}$ とし、各センサノードには初期位置が与えられる。センサノードは、音波によって通信を行うことが可能であり、その音波の最大通信距離は R [m] である。また、センサノードのセンシングデータ量は $Data$ [bit] とし、通信伝搬速度は V [m/s]、単位時間あたりのデータ送信量は B [bps] である。

3.3 タイムスロット

あるノードが複数のノードからデータを同時に受信したときにパケットが衝突する。また、あるノード自身が送信していても衝突をする。このパケットの衝突を避けるために TDMA (時分割通信) を利用するが [12]、水中の伝搬遅延により通信と受信の間にタイムラグが生じる。そこで各ノードに対して“通信してよい”時間枠 (以降、タイムスロット) を与え、衝突しないようなスケジュールを与える。センサノードの通信スケジューリングのために、1 つが W [s] の大きさのタイムスロットを用いる。タイムスロットの割当て例を図 2 に示す。ノード s_5 が $t = 1$ に送信すると、 s_2 と s_4 にはタイムラグが発生するために $t = 4$ に受信する。

3.4 インスタンスの例

ここでは、センサノード 5 つ、基地局ノード 1 つのインスタンスの具体例を示し、本問題について解説する。図 1 に示すような配置のセンサノード $s_0, s_1, s_2, s_3, s_4, s_5$ を考える。各ノード $s_0, s_1, s_2, s_3, s_4, s_5$ には、初期位置が与えられ、図 1 における各データ収集経路の距離は 5,000 [m] とする。また、センシングデータ $Data$ は 1,024 [bit]、最大通信距離 R は 5,000 [m]、単位時間あたりのデータ送信量 B は 512 [bps] とする。また、タイムスロット W は 1 [s] ごとに 1 つ割り当てる。通信伝搬速度は、 $V = 1,500$ [m/s] とする。このインスタンスに対して、提案手法を適用することで図 2 に示すようなタイムスロットスケジュールを与え



図 2 タイムスロット割当ての例
Fig. 2 An example of timeslots assignments.

ることができる。なお、図2の横軸はタイムスロットの番号、縦軸はノード番号となっている。センシングデータは1,024 [bit]で、単位時間あたりのデータ送信量は512 [bps]であるから、データを送信するのに2 [s]必要となり、1つの通信に対してタイムスロットは2つ割り当てられる。データは集約できるとしてあり、どの通信でもこの数は同じである。この例では、すべてのノードが基地局へデータを送信した際のタイムスロット数が22である。この数を最小化するようなデータ通信スケジュールを求めることが本問題の目的である。タイムスロットスケジュールは、衝突が発生しないようにスケジューリングしなければならない。

3.5 問題定義

本問題の入力として、基地局ノード B_s の位置 $B_s.pos$ 、センサノード集合 S 、センサノード $s \in S$ の位置 $s.pos$ 、センシングデータ $s.data$ 、データ量 $Data$ [bit]、通信距離 R [m]、データ伝送速度 B [bps]、伝搬速度 V [m/s]、タイムスロット1つの大きさ W [s] を与える。

出力は、全ノードにおけるすべての“送信”の集合であり、これを“スケジュール”とする。これについて以下で述べる。まず、各ノードが時刻を決めてデータを1回送信することを“送信”とする。“送信”は以下の式で表される。

$$send(s_i, s_j, D, [t_{start}, t_{end}]) \quad (1)$$

ここで、 s_i は送信元ノード、 s_j は送信先ノード、 D はこのとき送信するデータ集合とする。 t_{start} はノード s_i からノード s_j へ送信を開始するタイムスロット番号であり、 t_{end} をノード s_i からノード s_j へ送信を終了するタイムスロット番号とする。

図2のAの部分について説明する。このAの太い枠で囲まれた部分が、ノード s_4 がタイムスロット8~9にかけてデータ $s_4.data$ を送信することを示しており、また右上への矢印がノード s_4 からノード s_2 に向けた送信であることを示している。この送信はノード s_2 にデータを送るためのものであるが、通信範囲内にあるため、ノード s_1, s_3, s_5 にも通信が届いている。この例では偶然通信のタイミングが同じであるが、距離によっては同じタイムスロット番号とはならないことに注意せよ。なお、Aの部分の通信は、 $send(s_4, s_2, \{s_4.data\}, [8, 9])$ となる。以上のように、“送信”は表される。出力となる、全ノードにおけるすべての“送信”の集合である“スケジュール”は、以下の式で表される。

$$Schedule = \bigcup send(s_i, s_j, D, [t_{start}, t_{end}]) \quad (2)$$

目的関数は、すべてのセンシングデータが集約されるまでのタイムスロット数を最小化することであり、これについて以下で述べる。まず、ノード s_i のセンシングデータを $s_i.data$ とし、ノード s_i に時刻 t において集約されてい

るデータ集合 D_t^i とする。さらに、特定でないノード s のセンシングデータを $s.data$ とする。 D_t^i は以下の式で表される。

$$D_t^i = \{s.data | send(s_j, s_i, D, [t_{start}, t_{end}]), s.data \in D, t > t_{end} + [delaytime(s_j, s_i)]\} \quad (3)$$

ここで、 $delaytime(s_j, s_i)$ はノード s_i とノード s_j との通信に発生する遅延時間とし、以下の式で表される。 $delaytime(s_j, s_i) = distance(s_i, s_j)/V$ 。ここで、 $distance(s_i, s_j)$ はノード s_i とノード s_j 間の距離を返す関数である。

t_{finish} は、基地局ノード B_s にすべてのノードのセンシングデータが集約されるまでにかかるタイムスロット数であり、以下の式で与えられる。

$$t_{finish} = \min_{t \in \{t | D_t^{B_s} = \{s_1.data, s_2.data, \dots, s_n.data\}\}}(t) \quad (4)$$

この問題の目的関数は、以下の式で表される。

$$\text{minimize } t_{finish} \quad (5)$$

本問題の制約条件について以下に述べる。まず、ノード s_i がタイムスロット t_{start} からデータ集合を送信するとき、送信するデータはタイムスロット t_{start} までにそのノード s_i に集約されたデータ集合・式(3)に含まれていなければならない。このときのデータ集合を $D_{t_{start}}^i$ とする。この制約は、通信 $send(s_i, s_j, D, [t_{start}, t_{end}])$ に対して以下の式で表される。

$$D \subseteq D_{t_{start}}^i \quad (6)$$

次に、タイムスロットを割り当てる際、送信を終了するタイムスロット番号 t_{end} における制約は以下の式で表される。

$$t_{end} \geq t_{start} + [(Data/B)]/W \quad (7)$$

ここで、収集するセンシングデータは平均温度などを想定し、複数のセンシングデータであっても集約して同じデータ量に収まると仮定する。

また、ノード s_i からノード s_j へデータを送信する際、互いのノードは通信範囲内になければならない。この制約は、通信 $send(s_i, s_j, D, [t_{start}, t_{end}])$ に対して以下の式で表される。

$$distance(s_i, s_j) < R \quad (8)$$

あるノードが通信するとき、他の通信と衝突してはならない。以下の式(9)、(10)で表される通信について考える。

$$send(s_i, s_j, D, [t_{start1}, t_{end1}]) \quad (9)$$

$$send(s_k, s_l, D, [t_{start2}, t_{end2}]) \quad (10)$$

ここで、 t_{start1} は、ノード s_k からノード s_j へ送信を開始

するタイムスロット番号であり、 t_{end1} は、ノード s_k からノード s_j へ送信を終了するタイムスロット番号である。また、 t_{start2} は、ノード s_k からノード s_j へ送信を開始するタイムスロット番号であり、 t_{end2} は、ノード s_k からノード s_j へ送信を終了するタイムスロット番号である。

同じノードから送信する際 ($s_i = s_k$ のとき) の制約は以下のとおりである。

$$t_{start1} > t_{end2} \text{ OR } t_{end1} < t_{start2} \quad (11)$$

同じノードへと送信する際 ($s_j = s_l$ のとき) の制約は以下のとおりである。

$$t_{start1} + [\text{delaytime}(s_i, s_j)] > t_{end2} + [\text{delaytime}(s_k, s_l)]$$

OR

$$t_{end1} + [\text{delaytime}(s_i, s_j)] < t_{start2} + [\text{delaytime}(s_k, s_l)] \quad (12)$$

ある送信しようとするノードへ他のノードが送信する際 ($s_i = s_l$ のとき) の制約は以下のとおりである。

$$t_{start1} > t_{end2} + [\text{delaytime}(s_k, s_l)] \text{ or } t_{end1} < t_{start2} + [\text{delaytime}(s_k, s_l)] \quad (13)$$

最後に、制約条件の式を以下にまとめる。

subject to (6), (7), (8), (11), (12) and (13). (14)

4. 提案手法

4.1 概要

本章では提案手法である、水中ワイヤレスセンサネットワークのための伝搬遅延を考慮したタイムスロットスケジューリング手法について述べる。提案手法は、GAによって通信における優先度を割り当て、データ収集木の考慮による優先度の更新とパケットの衝突の可能性を考慮したタイムスロットの割当てのように3つの段階に分ける手法となっている。手順は以下のとおりである。まず、優先度割当てでは、優先度がGAの優先度の染色体として与えられ、優先度の数値が低い方から通信をしていく。次に、優先度更新では、データ収集木を考慮し、基地局へデータを効率良く収集するように優先度を並べ替え、更新する手順をとる。最後に、タイムスロット割当てでは、優先度順に衝突が起きないようにタイムスロット数が最小の位置で通信処理時間(送信・受信)を割り当てる。以上のように、提案手法では優先度によりセンサノードの通信を順序付け、優先度とデータ収集木より優先度を更新し、遺伝的アルゴリズムによって効率良くタイムスロットを割り当てていく点に特徴がある。

4.2 遺伝的アルゴリズム

4.2.1 概要

遺伝的アルゴリズム (GA: Genetic Algorithm) は、ダー

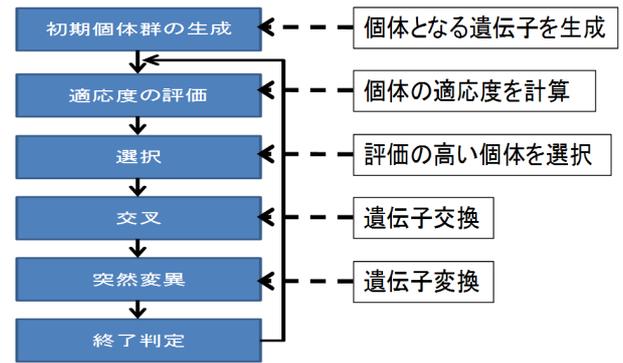


図3 遺伝的アルゴリズムのフローチャート

Fig. 3 Flowchart of genetic algorithm.

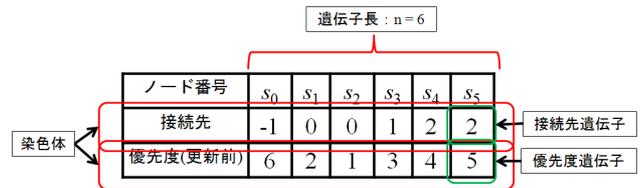


図4 遺伝子型での符号化

Fig. 4 Gene coding.

ウィンの進化論に基づく生物界の「遺伝的法則」を模倣した組合せ最適化アルゴリズムの1つである。アルゴリズムのフローチャートを図3に示す。

4.2.2 遺伝子の符号化

提案手法では、図4に示すようにデータ収集木の木構造(接続先)、各ノードの通信の優先度(優先度(更新前))をそれぞれの染色体とする。

まず、接続先の染色体(図4上部)が、データ収集木を表す。接続先の染色体の遺伝子長は n であり、これはノード数に等しい。接続先の染色体における各遺伝子は、各ノードのデータ送信先ノードの番号を示している。接続先遺伝子は、 n をノード数とし、 -1 もしくは $0 \sim n-1$ の $n+1$ 進法である。ただし、基地局ノードの遺伝子は -1 に固定する。たとえば、図4のノード s_1 における接続先は 0 であるため、ノード s_0 がデータ送信先であることが分かる。また、ノード s_0 における接続先は -1 であるため、データ送信先はない。他のノードに対しても同様であり、図4の遺伝子から図1で表されたデータ収集木を作成する。

また、優先度(更新前)の染色体は(図4下部)、各ノードの通信の優先度を表す。優先度(更新前)の染色体の遺伝子長は n であり、これはノード数に等しい。優先度(更新前)の染色体における各遺伝子は、通信の優先度を表し、この値の小さい方が優先的に通信できることを意味している。優先度(更新前)遺伝子は、 $1 \sim n$ の n 進法である。たとえば、図4でノード s_1 とノード s_2 ではそれぞれの優先度が 2 と 1 であるから、ノード s_2 に優先してタイムスロットが割り当てられる(4.4節で後述)。

ただしこの割当ては、データ収集木の構造を考慮して優

先度が更新 (4.3 節で後述) された後に行われることに注意せよ。以上の2つの染色体によって、タイムスロットスケジュールは一意に決まる。

4.2.3 初期個体群の生成

データ収集木の構築：初期個体群の接続先部分の初期化方法について述べる。以下のアルゴリズムを用いてデータ収集木をランダムに生成し、各個体に対して与える。

- (1) 基地局ノード s_0 の通信範囲内にあるノードをランダムに1つ選び、ノード s_i とする。
- (2) $V_T \leftarrow \{s_0, s_i\}$, $E_T \leftarrow \{e_{s_0, s_i}\}$ から初期木 $T = (V_T, E_T)$ を構成。ただし、 e_{s_i, s_j} はノード s_i とノード s_j 間の枝。連結候補のノード集合を $U \leftarrow V_T$ 。未接続のノード集合を $U' \leftarrow S \setminus V_T$ 。ただし、 S はノード全体の集合。
- (3) ($U = \emptyset$) かつ ($U' \neq \emptyset$) ならば、エラー (後述) としてアルゴリズム終了。
- (4) V_T からノードをランダムに1つ選び、 $s_j \in V_T$ とする。 $U \leftarrow U \cup \{s_j\}$ 。
- (5) 未接続のノード集合 U' 内に、 s_j の通信範囲内のノードが1つも存在しなければ、ステップ(4)へ。
- (6) ノード s_j の通信範囲内にあるノードをランダムに1つ選び、 $s_k \in U'$ とする。
- (7) $V_T \leftarrow V_T \cup \{s_k\}$, $E_T \leftarrow E_T \cup \{e_{s_j, s_k}\}$, $U' \leftarrow U' \setminus \{s_k\}$, $U \leftarrow V_T$ 。
- (8) $U' = \emptyset$ ならば、 $T = (V_T, E_T)$ を返り値としてアルゴリズム終了。
- (9) ステップ(4)へ。

このアルゴリズムでのエラーは、基地局から連結できないノードが存在することを示す。アルゴリズムがエラーとして終了した場合、すべてのノードでデータ収集木を構築できないためアルゴリズム全体を終了とする。

優先度 (更新前)：ここでは、優先度 (更新前) 部分の初期化方法について述べる。優先度 (更新前) として、巡回セールスマン問題と同様に各ノードごとに優先度が重複しないようランダムに順列を与える。

4.2.4 交叉

選択した2つの個体間において染色体の一部を交換して新しい個体を生成する操作である。ただし、接続先での交叉は考慮せず、優先度のみ交叉を適用する。つまり、交叉する2個体から優先度の染色体 (図4の下部分) をそれぞれ取り出し、図5に示す交叉を行う。接続先の染色体 (図4の上部分) は交叉しない。

優先度部分の交叉：優先度の染色体では、巡回セールスマン問題において巡回する利潤を順序付けることと似ているため、同様の交叉を用いることができる。ここでは、PMX法を用いる [13]。アルゴリズムは以下のとおりである。

- (1) 個体群 P から交叉する2個体を i, j とする (ただし、 $i, j \in P$)。

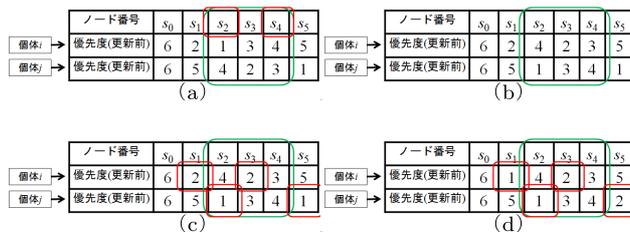


図5 交叉の例

Fig. 5 An example of crossover.

- (2) ランダムに2つ選ぶノードを s_k, s_l とする (ただし、 $k < l$)。
- (3) 個体 i の優先度の染色体 p に対して、ノード s_k の優先度遺伝子を $g_{i,k}^p$ 、ノード s_l の優先度遺伝子を $g_{i,l}^p$ とする。同様に、個体 j の優先度の染色体 p に対して、ノード s_k の優先度遺伝子を $g_{j,k}^p$ 、ノード s_l の優先度遺伝子を $g_{j,l}^p$ とし、 $g_{i,k}^p \sim g_{i,l}^p$ と $g_{j,k}^p \sim g_{j,l}^p$ を入れ替える (図5(a)~(b)参照)。つまり、通常の2点交叉を行う。
- (4) 個体 i, j の優先度の染色体 p において、重複した優先度遺伝子が存在しなければ、個体 i, j を返り値としてアルゴリズム終了。
- (5) そうでなければ、重複してる優先度遺伝子を重複ないように入れ替える (図5(c)~(d)参照)。入れ替えた個体 i, j を返り値としてアルゴリズム終了。

図5の交叉の例では、まず、交叉する2個体を i と j とし、ランダムに2つ選ぶノードを s_2, s_4 とする (図5(a)参照)。次に、優先度遺伝子 $g_{i,2}^p \sim g_{i,4}^p$ と優先度遺伝子 $g_{j,2}^p \sim g_{j,4}^p$ を入れ替える (図5(b)参照)。最後に、個体 i において $g_{i,1}^p$ と $g_{i,3}^p$ 、個体 j において $g_{j,2}^p$ と $g_{j,5}^p$ で重複した優先度遺伝子が存在しているので (図5(c)参照)、重複しないように $g_{i,1}^p$ と $g_{i,5}^p$ を入れ替える (図5(d)参照)。これらの個体 i, j を返り値とする。

4.2.5 突然変異

接続先の染色体 (図4の上部分) には“接続先部分の突然変異”を、優先度の染色体 (図4の下部分) には“優先度部分の突然変異”をそれぞれ適用する。

接続先部分の突然変異：接続先の染色体では、基地局ノード以外からランダムに選択したセンサノードの接続先を他のセンサノードにつなぎかえる方法を用いる。アルゴリズムは以下のとおりである。

- (1) 基地局ノード以外からランダムにノードを選び、ノード s_i とする。
- (2) 選択したノード s_i の接続先ノード s_j との辺 e_{s_i, s_j} を削除 ($E_T \leftarrow E_T \setminus \{e_{s_i, s_j}\}$) し、ノード s_i が存在する部分木を T_1 と、 s_j が存在する部分木 T_2 に分割。 T_2 のノード集合をコピーした集合 $U \leftarrow V_{T_2}$ 。
- (3) U からランダムにノードを選び、ノード s_k とする。 $U \leftarrow U \setminus \{s_k\}$ 。

表 1 更新後の優先度

Table 1 Updated priority values.

ノード番号	s_0	s_1	s_2	s_3	s_4	s_5
優先度 (更新後)	6	5	4	2	3	1

(4) ノード s_k がノード s_i の通信範囲内であれば, $E_T \leftarrow E_T \cup \{e_{s_i, s_k}\}$ とし, ステップ(6)へ.

(5) そうでなければ, ステップ(3)へ.

(6) $T = (V_T, E_T)$ を返り値としてアルゴリズム終了.

優先度部分の突然変異: 優先度の染色体では, 交叉のとき同様に巡回セールスマン問題で利用される突然変異と同様の2都市間順序入れ替えを用いる. アルゴリズムは以下のとおりである.

(1) 個体群 P から突然変異する1個体を i とする (ただし, $i \in P$).

(2) ランダムに選ぶノードを s_k と s_l とする.

(3) 優先度遺伝子 $g^{pi, k}$ と $g^{pi, l}$ を入れ替える.

(4) 個体 i を返り値としてアルゴリズム終了.

4.2.6 選択と終了判定について

提案手法では, 各世代において個体群と同数の子個体を生成し, 旧個体をすべて破棄する世代交代モデルを用いている. 選択にはルーレット選択を用いる. また, 終了判定は最終世代パラメータを用い, この世代に達すると終了とする [13].

4.2.7 適応度の計算

優先度を更新し (4.3 節で後述), タイムスロットを割り当てる (4.4 節で後述). そのときのすべてのセンシングデータが集約されるまでのタイムスロット数 t_{finish} が適応度となる.

4.3 優先度の更新

この節では, 染色体として与えられた優先度 (更新前) を, “子ノードからのデータを受け取らない状態で親ノードへデータを送信しない” という制約 (3.2 節参照) を満たすように優先度を更新するアルゴリズムについて述べる. 各センサノードには優先度が割り当てられ, 図 1 に示すようなデータ収集木を基に, 表 1 のように更新する. 優先度の更新方法は, 以下の手順のとおりである.

(1) 初期状態として, 子ノードのいないノード集合を S' . 更新候補のノード集合を $Q \leftarrow S'$. 更新状態のノード集合を $Q' \leftarrow \emptyset$. はじめに更新する優先度を $p \leftarrow 1$.

(2) 更新候補のノード集合 Q 内から最も優先度の値が小さい (より優先される) ノード s_i を選び, 優先度を p に更新. $Q \leftarrow Q \setminus \{s_i\}$. $Q' \leftarrow Q' \cup \{s_i\}$. $p \leftarrow p + 1$.

(3) ステップ(2) で選んだノード s_i に親ノード s_j があり, ノード s_j の子ノードすべてが Q' の要素であれば, $Q \leftarrow Q \cup \{s_j\}$ とし, ステップ(5)へ.

(4) そうでなければ, ステップ(2)へ.

(5) $Q = \emptyset$ であれば, アルゴリズム終了.

(6) そうでなければ, ステップ(2)へ.

4.4 タイムスロット割当て

4.3 節で更新した優先度を用いて, 各ノードに通信 (送信・受信) するタイムスロットを割り当てる方法を割り当てる. 割当て方法は, 以下の手順のとおりである. 割り当てる時刻を t とする.

(1) $Schedule \leftarrow \emptyset$

(2) 優先度 $p \leftarrow 1$ とし, 優先度が p のノードを選び, s_i とする.

(3) $Schedule \leftarrow \{send(s_i, s_j, D, [t, t + \lfloor ([Data/B]) / W \rfloor])\} \cup Schedule$. ただし, ノード s_j は, 染色体として与えられたノード s_i の接続先である (4.2.2 項および図 4 参照).

(4) $Schedule$ が制約条件 3.5 節式 (14) を満たすならば, ステップ(6)へ.

(5) そうでなければ,

$Schedule \leftarrow Schedule \setminus \{send(s_i, s_j, D, [t, t + \lfloor ([Data/B]) / W \rfloor])\}$, $t \leftarrow t + 1$ とし, ステップ(3)へ.

(6) $p \leftarrow p + 1$.

(7) $p < n + 1$ であれば, ステップ(2)へ.

(8) そうでなければ, $Schedule$ を出力し, アルゴリズム終了.

5. 実験

5.1 実験概要

提案手法の性能を評価するため, 得られたスケジュールのタイムスロット数を計測した. 比較対象として, Chenらが提案している手法 (データ分割法) [15] および提案手法の GA 部分をランダム探索に置き換えた手法 (ランダム法) を用いた. また実験環境として, OS は Microsoft Windows 7 Professional Service Pack 1 を, コンパイラは gcc version 4.5.3 を用いた.

5.2 実験パラメータ

本実験によるパラメータは以下のとおりである. 実験の機材のモデルとして, LinkQuest 社 [14] の UWM3000H モデルを想定する.

- センシングデータ $D = 1,024$ [bit]
- 最大通信距離 $R = 5,000$ [m] (参考: 文献 [14])
- 単位時間あたりのデータ送信量 $B = 320$ [bps] (参考: 文献 [14])
- 単位時間あたりのタイムスロット数 $t = 1$
- 通信伝搬速度は, $V = 1,500$ [m/s]

5.3 比較手法

5.3.1 データ分割法

2つ目の比較手法のために、Chen らの手法 [15] を利用する。この手法は、UWSN において 2 ホップ先のノードに少ない遅延でデータを送信するものである。送信元となるノードは、送信するデータを複数の部分データに分割する。これら部分データを、送信元から 1 ホップの中継ノードに送信する。中継ノードは、部分データの受け取りが終了してから送信先ノードにデータを送信する。この手法は、2 ホップごとにこれを繰り返すことにより 2 ホップ以上離れた送信先ノードへもデータを送信することができる。中継ノードはデータを受信してから送信するため、データの大きさに比例する通信遅延時間が必要となる。この手法は、データを分割することにより、この通信遅延時間を減少させる点に特色がある。

この手法は 2 ノード間の通信を仮定しているため、全センサノードに対してこの手法を繰り返し用いるものをデータ分割法と題して比較のために用いる。提案手法では送信データの分割を想定していないが、データ分割法では分割できるとする。また、2 ホップごとにデータが集まった際には、提案手法と同様に、受信したデータとそのセンサのデータを集約してデータを送信できるとする。中継ノードは、送信元ノードのデータのみを送ることができるとする。ただし、中間ノードで一度送信元データが揃った場合は、集約されたものと見なし自身のノードを合わせて集約し送信可能とする。

また、Chen らの手法はデータ送信の手法であるため、データ収集の手法とするため以下の改変を加える。全センサノードのデータを収集できるよう、逐次的に Chen らの手法を繰り返すこととする。

5.4 実験：提案手法の性能評価

5.4.1 ノード数の変化によるタイムスロット数の評価

フィールド 1 辺の大きさを 25,000 [m] に固定し、ノード数を 10 から 50 まで変化させて実験を行った。実験結果を図 6 に示す。この結果は 30 回試行の平均である。

ノード数 50 のときに、データ分割法より提案手法が 58%程度少ないタイムスロット数のスケジュールを得ていることが分かる。このことから、遅延時間の隙間に他の通信を割り当てる提案手法が有効であると考えられる。

また、ランダム法より提案手法が 15%程度少ないタイムスロット数のスケジュールを得ていることが分かる。あまり大きな差ではないが、GA を用いてデータ収集経路やタイムスロットの組合せを得ることがある程度有効であると考えられる。

ノード数が多くなるにつれタイムスロット数が増加するのは、送信するデータが増加し、データの衝突を避けるために必要とするタイムスロット数が増加するからだと思

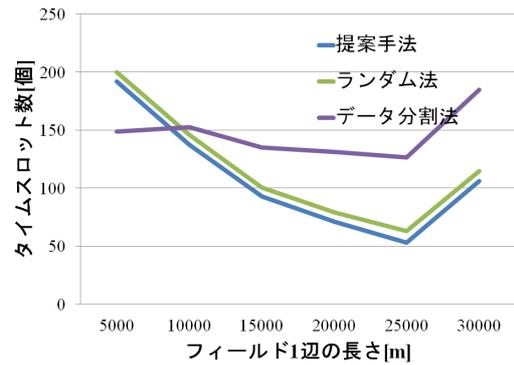


図 6 ノード数の変化に対する評価

Fig. 6 The number of timeslots according to the number of nodes.

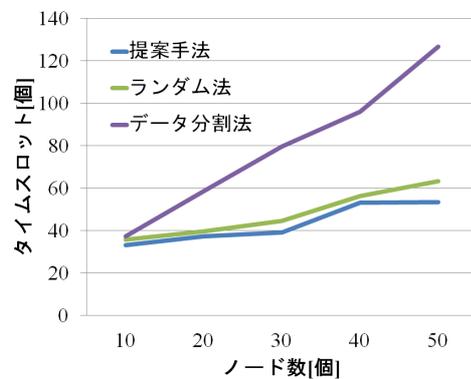


図 7 フィールドの大きさの変化に対する評価

Fig. 7 The number of timeslots according to field size.

られる。

5.4.2 フィールドの大きさの変化によるタイムスロット数の評価

ノード数を 50 個に固定し、フィールド 1 辺の長さを 5,000 [m]~30,000 [m] で変化させて実験を行った。実験結果を図 7 に示す。この結果は 30 回試行の平均である。なお、基地局ノードはフィールドの中心に配置した。

フィールドの大きさが 25,000 [m] のとき、データ分割法より提案手法が 58%、ランダム法より提案手法が 26%程度少ないタイムスロット数のスケジュールを得ていることが分かる。

フィールドが狭いときタイムスロット数が大きいのは、通信範囲内にノードがより多く存在すると衝突を起こしやすくタイムスロットをより多く消費してしまうからであると考えられる。

また、必要なタイムスロット数の最小の理論値は、フィールド 1 辺の長さ、音速、通信時間、基地局からフィールド端までの最小ホップ数で求まる。たとえば、フィールド 1 辺の長さが 30,000 [m] であれば、基地局ノードがフィールド中心にあるとして $15,000 [m] \div 1,500 [m/s] \approx 10 [s]$ となり、タイムスロットは 10 である。また、通信時間を 2 とす

表 2 各手法の収束までの評価回数とそのときの実行時間

Table 2 Number of evaluation and execution time.

	評価回数 [回]	計算時間 [s]	解 [タイムスロット数]
提案手法	100,000	22.51827	53.53333
ランダム法	100,000	37.00987	63.16667
データ分割法	-	0.015	126.6

ると、最小ホップ数は3であるから、必要なタイムスロット数の最小の理論値は $10 + 3 \times 2 = 16$ である。このことから、フィールドの大きさがさらに大きくなった場合は、必要タイムスロット数はむしろ増大していくと考えられる。

さらに、データ分割法では実験結果から分かるように、ノードの配置が密な場合に有用である。一方、提案手法では、ノードの配置が疎な場合に有用であるため、これら2つを統合した手法を開発できれば有用であると考えられる。

5.4.3 実行時間における比較実験

表 2 に、50 ノードを持つインスタンスに対する各手法での評価回数とそのときの計算時間および解を示す。データ分割法は発見的な手法でないため、評価回数を示していない。22 秒程度であれば、ノードが固定されて同じスケジュールを何度も利用できる場合は十分利用できると思われる。

6. おわりに

本研究では、GA を用い通信における優先度を決定し、データ収集経路と優先順位から伝搬遅延を考慮した水中ワイヤレスセンサネットワーク (UWSN) のための効率的なタイムスロットスケジューリングを提案する。本研究では、GA によって通信における優先度を割り当て、データ収集経路の考慮による優先度の更新とパケットの衝突の可能性を考慮したタイムスロットの割当てを3段階に分けている点に特色がある。

提案手法によって得られたタイムスロットスケジューリングの評価を行うために、比較手法としてデータ分割法と比較した。その結果、58%程度少ないタイムスロットスケジュールを得ることができた。このことから提案手法が伝搬遅延を考慮した水中ワイヤレスセンサネットワークのタイムスロットスケジューリングに有効であることが分かった。

今後の課題としては、データ分割法と提案手法との統合、タイムスロットの大きさの影響、音波の合成による影響、ドップラー効果による影響について調査する予定である。また、遅延時間の不連続性がタイムスロット方式と相性が悪いため、タイムスロットを用いない方法への拡張などがあげられる。

参考文献

[1] Wan, C.-Y., Campbell, A.T. and Krishnamurthy, L.: PSFQ: A Reliable Transport Protocol for Wireless Sen-

- sor Networks, *WSNA* (2002).
- [2] Wan, C.-Y., Eisenman, S.B. and Campbell, A.T.: CODA: Congestion Detection and Avoidance in Sensor Networks, *ACM SenSys* (2003).
- [3] Warneke, B., Last, M., Liebowitz, B. and Pister, K.S.J.: Smart Dust: Communicating with a Cubic-Millimeter Computer, *IEEE Computer*, pp.44-51 (2001).
- [4] クロスボー株式会社製品カタログ, 入手先 (<http://www.xbow.jp/motecatalog.pdf>).
- [5] Lucero, S.: 産業オートメーションの無線センサーネットワーク (WSN): 監視と管理アプリケーションの市場, ABI Research (2007).
- [6] 大林成行: アドホックタイプのセンサネットワークを導入した空間データの新しい収集・管理・提供方法の開発, 財団法人日本建設情報総合センター (JACIC) 研究助成事業報告書 (2005).
- [7] 平井一弘, 里見知昭, 酒匂一成, 深川良一: 無線センサネットワークの斜面防災への適用に関する実験的研究, 歴史都市防災論文集, Vol.3, pp.99-104 (2009).
- [8] Cui, J.H., Kong, J., Gerla, M. and Zhou, S.: Challenges: Building Scalable Mobile Underwater Wireless Sensor Networks for Aquatic Applications, UCONN CSE tech. repo. UbiNET-TR05-02 (2005).
- [9] Akyildiz, I.F., Pompili, D. and Melodia, T.: Challenges for Efficient Communication in Underwater Acoustic Sensor Networks, *ACM SIGBED Review*, Vol.1, No.1 (2004).
- [10] Lu, C., Wang, S. and Tan, M.: A time synchronization method for Underwater Wireless Sensor Networks, *Control and Decision Conference IEEE*, pp.4305-4310 (2009).
- [11] Liu, L.: Time Synchronization of Underwater Wireless Sensor Networks, *Smart Wireless Sensor Networks*, Tan, Y.K. (Ed.), ISBN: 978-953-307-261-6, InTech (2010), available from (http://www.intechopen.com/source/pdfs/12456/InTech-Time_synchronization_of_underwater_wireless_sensor_networks.pdf).
- [12] Che, X., Wells, I., Dickers, G. and Kear, P.: *Frame Design for a Prototype Underwater RF Electromagnetic Communication Sensor System*, ASTEC Internal Publications, available from (<http://www.fishnetz.co.uk/Publications.html>).
- [13] 棟朝雅晴: 遺伝的アルゴリズム—その理論と先端的手法, 森北出版株式会社 (2008).
- [14] LinkQuestInc, available from (<http://www.link-quest.com/index.htm>).
- [15] Chen, Y.-S., Juang, T.-Y., Lin, Y.-W. and Tsai, I.-C.: A Low Propagation Delay Multi-Path Routing Protocol for Underwater Sensor Networks, *Journal of Internet Technology*, Vol.11, No.2, pp.153-165 (2010).



浜田 龍平

昭和 63 年生。平成 23 年広島市立大学情報科学部システム工学科卒業。平成 25 年広島市立大学大学院情報科学研究科システム工学専攻修士課程修了見込み。現在、同大学院にて遺伝的アルゴリズムの研究に従事。



村田 佳洋 (正会員)

平成 12 年奈良先端科学技術大学院大学情報科学研究科修士課程修了, 平成 15 年奈良先端科学技術大学院大学情報科学研究科博士後期課程修了, 平成 15 年 4 月より奈良先端科学技術大学院大学情報科学研究科助手. 平成 19

年 10 月より, 広島市立大学情報科学研究科准教授. 遺伝的アルゴリズムと観光ナビゲーション技術の研究に従事.