

An autonomous distributed algorithm based on calling behavior of frogs for a connected dominating set

Rei Tsutsui and Akihiro Fujiwara

Graduate School of Computer Science and System Engineering

Kyusyu Insutitute of Technology

Iizuka, Fukuoka, 820-8502, Japan

E-mail: m232052r@mail.kyutech.jp, fujiwara@cse.kyutech.ac.jp

Abstract—In the sensor network, a connected dominating set is a set of connected sensors such that every sensor not in the set is adjacent to at least one sensor in the set. The connected dominating set with a small number of sensors is desirable because the set reduces network energy and communication costs.

In this paper, we propose an autonomous distributed algorithm, which is based on calling behavior of frog, for constructing the connected dominating set. We implement the proposed algorithm in a simulation environment, and evaluate validity of the proposed algorithm.

Keywords: sensor network, connected dominating set, calling behavior of frogs

I. はじめに

センサネットワークでは、情報を収集するために、情報伝達の経路となるセンサ集合を求める必要があり、このセンサ集合の一つとして、連結支配集合がある。連結支配集合とは、全センサ集合に隣接するセンサの集合であり、さらに集合の任意のセンサと集合の他のセンサとが連結している場合、この集合を連結支配集合 [1] という。最小の連結支配集合を求める問題は NP 困難であり [2]、多くの近似解を求める方法が考案されている [3]。

本研究では、この連結支配集合に対して、蛙のサテライト行動に基づく自律分散型アルゴリズムの提案を行う。蛙には、周辺に多くの蛙が存在する場合、鳴くことを止め、体力を温存するという習性がある [4]。この習性を用いてセンサの休止、稼動を行うことにより、連結支配集合を構築する。提案アルゴリズムでは、周辺のセンサとのみ通信を行い、連結支配集合を構築することを目的とする。

また、提案アルゴリズムをシミュレーション環境に実装し、有用性の評価を行う。本シミュレーション結果より、蛙のサテライト行動を用いた提案アルゴリズムにより得られる解は、連結支配集合を構築することができた。さらに、提案アルゴリズムを改善することで、より大きい割合で連結支配集合を構築し、また、連結支配集合の数を削減することができた。

II. 準備

A. センサモデル

本研究の対象であるセンサネットワーク $G = (V, E)$ は、センサ集合 $V = \{I_1, I_2, \dots, I_n\}$ と通信可能なセンサ間の経

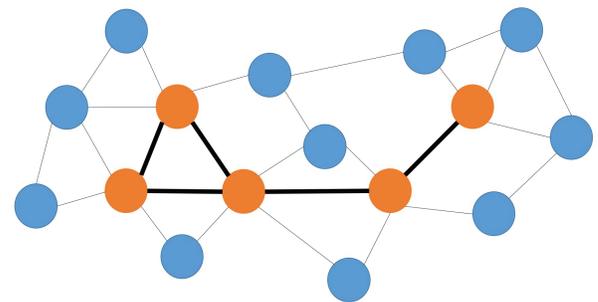


Fig. 1. 連結支配集合

路を表すリンク集合 E で定義される。センサ I_i とセンサ I_j が互いに通信可能であるとき、グラフ G 上の I_i, I_j 間に双方向通信リンク $e_{i,j} \in E$ が存在する。なお、このときのセンサ I_i と I_j は隣接しているという。また、この隣接するセンサ I_i と I_j との間のみ、メッセージの送受信が可能であるとする。さらに、通信衝突などによるメッセージの消失は起こらないものとする。

今回のセンサは隣接しているとき、双方向で通信を行えるものとする。また、センサは2つのモードを持ち、モード毎に機能が異なる。以下に、センサの各モードを説明する。

アクティブ: 通信領域内のセンサと通信を行い、メッセージを送信する。

スリープ: アクティブなセンサからメッセージを受信する。

B. 連結支配集合

センサ集合 $V = \{I_1, I_2, \dots, I_n\}$ とリンク集合 E で定義されるセンサネットワークを表すグラフ $G = (V, E)$ に対する連結支配集合とは、以下の3条件を満たす頂点の集合 V' である。

- 1) V' は V の部分集合である。
- 2) V' は連結である。
- 3) V' に含まれない全ての頂点は、 V' に含まれる隣接頂点をもつ。

Fig.1 に連結支配集合の例を示す。図のオレンジ色の円は連結支配集合に含まれるセンサを表しており、青色の円は連結支配集合に含まれない円を表している。

次に、本研究における連結支配集合の評価指標について説明する。本研究では、連結支配集合の構築アルゴリズムを提案するが、評価指標は以下の通りである。なお、本提案アルゴリズムはほとんど通信を行わない分散アルゴリズムであり、必ずしも連結支配集合が得られるわけではないことに注意が必要である。

評価指標：

- 出力として得られた連結支配集合のセンサ数
- 出力されたセンサ集合が連結支配集合の条件を満たす割合

III. 蛙のサテライト行動に基づく連結支配集合構築アルゴリズム

A. 蛙のサテライト行動

まず最初に、提案アルゴリズムの基本アイデアとなる蛙のサテライト行動について説明する。蛙のサテライト行動とは、蛙の発声行動における特徴の一つである。ある種の蛙には周囲に多くの雄が存在する時、鳴くことによって求愛を成功させる確率を上げる利益よりも、鳴くことによって消費する体力の損失のほうが大きいと判断し、鳴くことを止めるという特徴がある。また、他の雄と同時に鳴いては求愛対象の蛙に音を聞き分けてもらえないため、鳴くタイミングをずらすという特徴も持っている。

本研究ではセンサを蛙と見立て、これらの特徴をセンサネットワークに応用する。蛙が発声する場合、センサは周辺に刺激を与えるアクティブモードになり、蛙が鳴かずに待機している場合、センサは休止しスリープモードになる。これら2つのモードを蛙の特徴により切り替えることで、連結支配集合を構築できると考えられる。

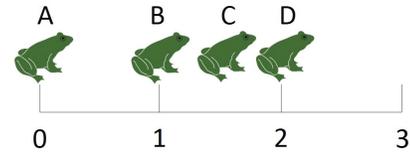
B. 蛙のサテライト行動に基づく既存アルゴリズム

本節では、前述の蛙のサテライト行動に基づく既存アルゴリズム [5] を説明する。このアルゴリズムは、初期段階でセンサに発声閾値 v と、ランダムな状態値 c 、状態値の進み幅 a を与える。各センサは以下の3フェーズを繰り返す。

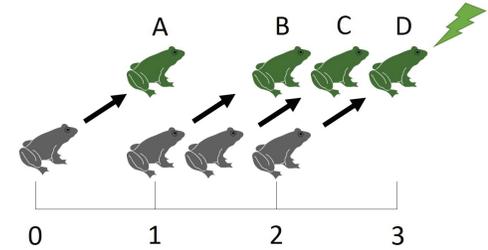
- フェーズ 1: 状態値 c が発声閾値 v に達している場合、センサはアクティブモードに移行し周囲のセンサに刺激を与える。
- フェーズ 2: アクティブモードになったセンサは状態値 c をリセットし、センサはスリープモードに移行する。
- フェーズ 3: アクティブモードにならなかったセンサは、1ステップの実行後に、各センサーの状態値 c に進み幅 a を加算する。

Fig.2の例を用いて、本アルゴリズムの説明を行う。Fig.2の(a)は初期状態の発声閾値、各センサの状態値を表す。このときの各センサの発声閾値 $v = 3$ とする。また、状態値の進み幅 a は v を2で割った値の整数部分である $a = 1$ とする。また、センサA、センサB、センサC、センサDの初期状態値をそれぞれ c_A, c_B, c_C, c_D とする。Fig.2の(a)の場合では、 $c_A=0, c_B=1, c_C=1.5, c_D=2$ より、状態値 c が発声閾値 v に達しているセンサが存在しないため、アクティブモードに移行するセンサは存在しない。

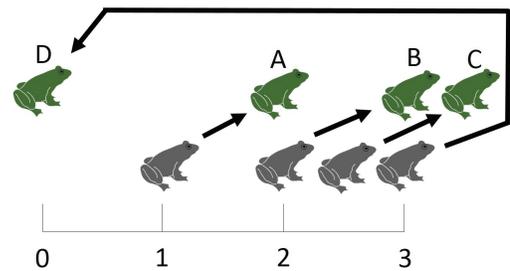
Fig.2の(b)は1ステップ後の状況を表す。このとき、各センサの状態値 c は更新され、 $c_A=0+1=1, c_B=1+1=2,$



(a) 初期状態



(b) アクティブモードへの移行



(c) スリープモードへの移行

Fig. 2. 既存アルゴリズムの動作例

$c_C=1.5+1=2.5, c_D=2+1=3$ となる。この時センサDの状態値 c_D が3となり、発声閾値 $v=3$ に達したためアクティブモードに移行している。

Fig.2の(c)はさらに1ステップ後の状況を表す。このとき、センサDの状態値 c_D が0にリセットされ、センサA、センサB、センサCの状態値が更新され、 $c_A=1+1=2, c_B=2+1=3, c_C=2.5+1=3.5$ となる。

IV. 提案アルゴリズム

本節では、提案手法による分散型連結支配集合構築を行うアルゴリズムを紹介する。本節で述べるアルゴリズムは、前節で説明したアルゴリズムに進み幅を調整する概念を組み込んだアルゴリズムである。各センサが、自身の状態値に応じてステップ数が増えた時の状態値の進み幅を調整することで、一定のタイミングでアクティブモードになることを目指す。このアルゴリズムは、以下の3つのフェーズで構成される。

- フェーズ 1: センサ s_v の状態値 c_v が発声閾値 v_v に達した場合、 s_v の隣接センサ群 S_N は各センサ $s_i (\in S_N)$ の状態値に応じた刺激を受け、 s_i の状態値の進み幅 a_i を更新する。

- フェーズ2: フェーズ1でアクティブモードになったセンサ s_v は、進み幅と状態値をリセットし、スリープモードに移行する。
- フェーズ3: 1ステップの実行後に、状態値の進み幅 a_i 分、各センサ s_i の状態値 c_i を更新する。

以下では、それぞれのフェーズの詳細について述べる。

フェーズ1: センサ s_v の状態値 c_v が、発声閾値 v_v に達した場合、隣接センサ群 S_N は各センサ $s_i (i \in S_N)$ の状態値に応じた刺激を受け、 s_i の進み幅 a_i を更新する。

あるセンサがアクティブモードとなる度に、周辺のセンサは自身の状態値に応じた刺激を受ける。発声閾値 v_v の半分の値を基準値 $\frac{v_v}{2}$ 、自身の状態値を c_i 、カウントの進み幅を a_i とする。刺激の大きさは、 $|\frac{v_v}{2} - c_i|$ に比例するものとし、 $\frac{v_v}{2} > c_i$ の場合、正の刺激を受け進み幅 a_i は大きくなり、以下の式で与えられるものとする。

$$a_i = a_i + \left| \frac{v_v}{2} - c_i \right|$$

一方、 $c_i > \frac{v_v}{2}$ の場合、負の刺激を受け進み幅 a_i は小さくなるものとし、以下の式で与えられるものとする。

$$a_i = a_i - \left| \frac{v_v}{2} - c_i \right|$$

$c_i = \frac{v_v}{2}$ の場合、正と負の刺激が打ち消しあい、進み幅 a_i には影響せず、以下の式で与えられるものとする。

$$a_i = a_i$$

ここで、センサ s_v の通信領域内にある自身を除くセンサの数を N_v とすると、発声閾値 $v_v = N_v$ となる。

フェーズ2: フェーズ1でアクティブモードになったセンサ s_v は、進み幅 a_v と状態値 c_v をリセットし、スリープモードに移行する。

アクティブモードになったセンサ s_v は、以下の式の通り、進み幅 a_v は初期状態の進み幅 a_r にリセットされ、状態値 c_v を0にリセットする。

$$a_v = a_r$$

$$c_v = 0$$

ただし、複数のセンサが同時にアクティブになった場合、以下の式に従って状態値をリセットする。状態値 c_v は確率変数 $x=0$ の時、0にリセットされ、 $x=1$ の時1にリセットされる。確率変数 $x=0$ になる確率 $P_0 = \frac{1}{2}$ で、 $x=1$ になる確率 $P_1 = \frac{1}{2}$ とする。

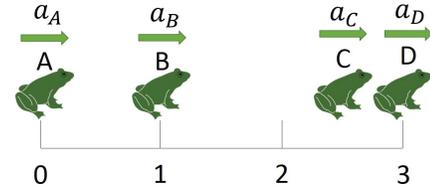
$$c_v = \begin{cases} 0 & (x=0) \\ 1 & (x=1) \end{cases}$$

フェーズ3: ステップ数が1増えた時に進み幅 a_i 分、状態値を更新する。

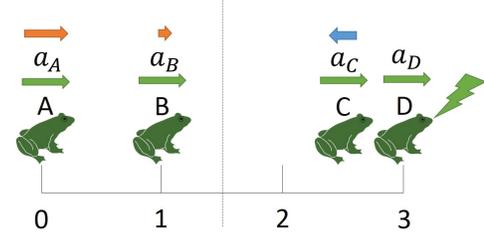
進み幅 a_i を状態値 c_i に加算する。更新後の状態値 c_i は、以下の式で与えられる。ただし、アクティブからスリープとなったセンサの状態値は更新しない。

$$c_i = c_i + a_i$$

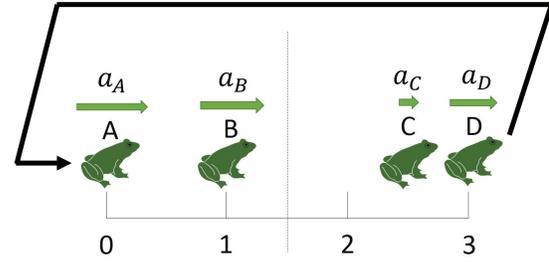
Fig.3 の例を用いて、本アイデアの説明を行う。Fig.3 の(a)は初期状態を示す。このとき、発声閾値 $v_v=3$ とし、状態値の進み幅 $a_i=1$ とする。センサ A、センサ B、センサ C、



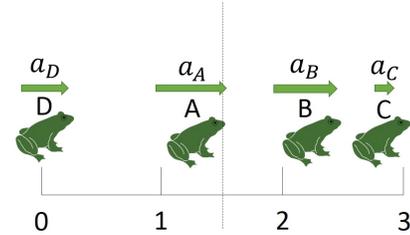
(a) 初期状態



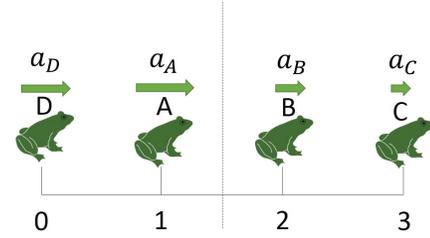
(b) アクティブモードへの移行



(c) スリープモードへの移行



(d) 1ステップ後の状態



(e) ステップ数が十分に経過

Fig. 3. 提案アルゴリズムの動作例

センサ D の初期状態値をそれぞれ c_A, c_B, c_C, c_D とすると、 $c_A=0, c_B=1, c_C=2.5, c_D=3$ である。Fig.3 の (b) ではフェーズ1での進み幅の更新の様子を示す。青い矢印は負の刺激を、赤い矢印は正の刺激を示し、矢印の長さは刺激の大きさを示す。中央の点線は基準値 $\frac{v_v}{2}$ を示す。状態値 $c_D=3$ となり、センサ D がアクティブモードとなったので、各セ

ンサは刺激を受ける。

Fig.3 の (c) ではセンサ D のスリープモードへの移行と、各センサの進み幅の更新を示す。各センサは更新された進み幅 a_i 分、状態値を更新する。

Fig.3 の (d) ではセンサ D の進み幅と状態値がリセットされ、各センサが状態値を更新した様子を示す。この時、発声閾値に達したセンサが存在しないため進み幅の更新は行われず、再び進み幅 a_i 分、状態値を更新する。

Fig.3 の (e) にステップ数が十分経過した状態を示す。進み幅を更新し、正と負の刺激が釣り合った時、センサは効率よくモードを移行できる状態になる。

V. 実験結果と考察

A. 既存手法との比較

提案手法をシミュレーション環境に実装し、評価実験を行った。なお、本実験は、LEDA[6] を用いて実装を行った。また、センサは 100×100 の矩形領域にランダムに配置されるものとし、各センサの通信半径は 10 として、実験を行った。Fig.4 に実験結果を示す。

Fig.4 より、提案手法が、既存手法よりも連結支配集合を構築する割合が大きいことがわかる。しかし、連結支配集合が必ず構築されるとは限らない。一方、連結支配集合の数は既存アルゴリズムの方が少ない。これは単純に、連結支配集合の数が増えたため、割合が大きくなったためだと考えられる。

連結支配集合が構築されないのは、センサが正方領域にランダムで配置された時に、センサが正方領域の縁に配置されると、その付近のセンサ集合が疎になるためであると考えられる。また、各センサがアクティブモードになる間隔が均等になっていない、ステップ数が少ない段階での割合を計算していたため、割合が小さくなったものと考えられる。

B. 提案手法の改善

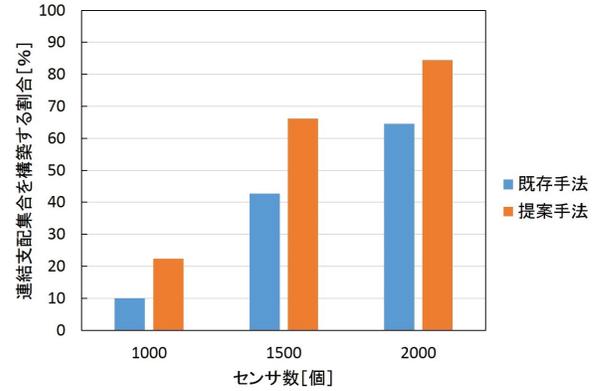
既存手法と提案手法の比較の実験結果より、2つの問題点があった。ここでは、その2つの問題点を改善して提案手法の評価実験を行った。

1つ目の改善点は、Fig.5 のように、センサの矩形領域の中にさらに小さい矩形領域を作り、その小さい矩形領域の中でのみ連結支配集合の条件を満たしているかを調べる。これにより、矩形領域の縁で連結支配集合の条件を満たさなくなるのを抑える。

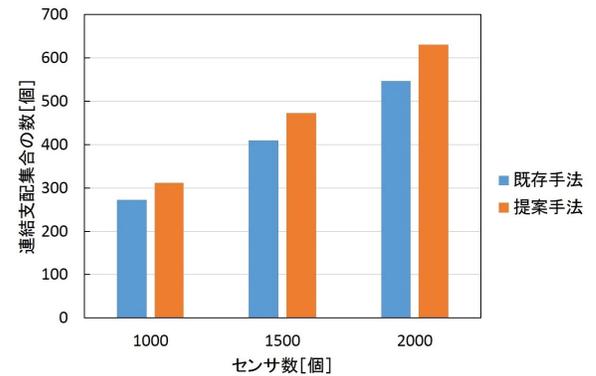
2つ目は、1つの試行で 1000 ステップを実行する際の最初の 100 ステップまでを準備期間とし、出力を行わないようにする。これにより、十分にステップ数が経過していない状態での結果を出力しないようにする。

また、提案手法のアルゴリズムにおいて、センサがアクティブになり、状態値が 0 へリセットされるが、この時の状態値を発声閾値に近づける。これにより、センサがよりアクティブになりやすく、連結支配集合を構築する割合が高くなる。

この3つの改善点をアルゴリズムに実装し、実験を行った。実験結果を TABLE I に示す。なお、ステップ数は 1000 ステップ、試行回数は 100 回とする。また、改善点に従って、1000 ステップ中の最初の 100 ステップは出力しないものとする。さらに、状態値のリセット値 $c_r = \frac{v}{3}$ とする。TABLE



(a) センサ数と連結支配集合を構築する割合



(b) センサ数と連結支配集合の数

Fig. 4. 既存手法と提案手法の比較

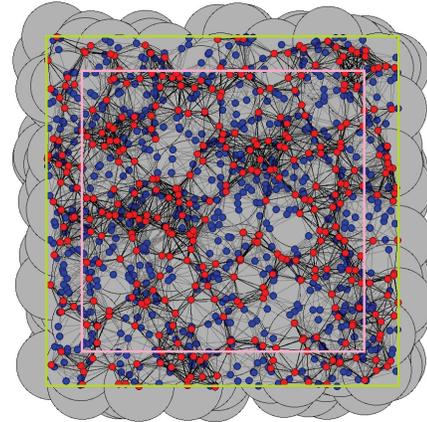


Fig. 5. 2つの矩形領域の例

I の CDS 割合は連結支配集合の条件を満たす割合、CDS 数は連結支配集合の数を示す。

TABLE I を見ると、改善前よりも高い数値が出ていることがわかる。また、通信領域の半径が小さくなると、アクティブとなるセンサの数が増えている。これにより、周囲にアクティブなセンサが少ないと、より多くのセンサがア

TABLE I
提案手法の改善結果

		全センサ数		
		1000	1500	2000
CDS 割合 (%)	改善前	22.3810	66.1680	84.5150
	改善後	94.3905	99.5994	99.5943
CDS 個数 (個)	is	311.589	472.393	630.000
	改善後	282.452	419.422	556.121

クティブとなることがわかる。まとめると、連結支配集合の条件を満たす割合は全センサ数と通信領域の半径に比例し、連結支配集合の数は、全センサ数に比例するが、通信領域の半径に反比例する。

この結果より、センサ数が少なかったり、通信領域の半径が小さすぎると、連結性を保てなくなることがわかる。そのため、適切なセンサ数と通信領域の半径を選択する必要がある。

VI. まとめ

本研究では、蛙の行動特性を用いることにより、連結支配集合を構築するアルゴリズムの提案を行った。既存手法と提案手法の比較実験を行い、見つかった問題点を改善して検証を行った。センサ周辺の環境設定によって結果が変わるため、適切な値を選択する必要がある。

今後の課題として、センサにバッテリーを仮定し、ランダムにアクティブとなるセンサとの寿命比較を行うことや、センサ集合が疎な場合でも連結支配集合が構築可能なアルゴリズムの提案が挙げられる。

REFERENCES

- [1] S. Kamei, H. Kakugawa. A Self-Stabilizing Distributed Approximation Algorithm for the Minimum Connected Dominating Set *International Journal of Foundations of Computer Science*, pp. 459-476, 2010.
- [2] R. Jovanovic, M. Tuba, D. Simian. Ant Colony Optimization Applied to Minimum Weight Dominating Set Problem *Proc. 12th WSEAS International Conference on AUTOMATIC CONTROL*, 2010.
- [3] Chi-Fu Huang, Yu-Chee Tseng. The coverage problem in a wireless sensor network *Mobile Networks and Applications*, Vol. 10, pp.519-528, 2005.
- [4] A. Mutazono, M. Sugano, M. Murata. Energy efficient self-organizing control for wireless sensor networks inspired by calling behavior of frogs. *International Journal of Sensor Networks*, 2012.
- [5] 横溝 康作, 藤原 暁宏 蛙のサテライト行動に基づくセンサカバーの構築. 卒業論文, 2015.
- [6] 浅野 哲夫, 小保方 幸次. LEDA で始める C/C++ プログラミング入門からコンピュータ・ジオメトリまで. サイエンス社, 2002.