

個体群プロトコルにおけるほぼ時間最適な 匿名任意グラフ上の緩安定リーダ選挙

金谷温貴*¹ 江口僚太¹ 笹田大翔¹ 井上美智子¹

¹ 奈良先端科学技術大学院大学

概要

個体群プロトコルモデルは Angluin らによって提案された、複数の個体が互いに通信しあって計算するモデルである。本研究では、任意の通信グラフ上において任意の初期状況から実行を開始し、ただ1つのユニークなリーダ個体を決めるリーダ選挙問題を扱う。しかしながら、任意グラフ上での自己安定リーダ選挙問題は正確な個体数を知らずに解くことができないと証明されているため、緩安定リーダ選挙問題に焦点を当てる。緩安定性とは、比較的短い時間で目的の状況（リーダ個体が1つの状況）へ収束したのち、比較的長い時間目的の状況を保持することである。任意グラフ上での緩安定リーダ選挙問題は Sudo らがすでに研究しており、個体が互いに異なる識別子を持っており、かつ正確な個体数 n の上界 N を与えた場合に、期待収束時間 $O(mN \log n)$ 、期待保持時間 $\Omega(Ne^{2N})$ 、メモリ使用量 $O(\log N)$ ビットのプロトコルを提案している。ただし、 m はグラフの辺の数を表す。また、Sudo らは期待保持時間 $\Omega(e^N)$ に必要な期待収束時間が $\Omega(mN)$ であることも示している。

本論文では、メモリの使用量を増やすことで、個体が互いに異なる識別子を持っているという仮定を必要とせずかつ収束時間が下界に近いプロトコルを提案する。具体的には、個体数の上界 N と個体の次数の最大値の上界 Δ を与えた場合に、期待収束時間 $O(mN \log n)$ 、期待保持時間 $\Omega(Ne^{2N})$ 、メモリ使用量 $O(\Delta \log N)$ ビットとなる乱択プロトコルと、期待収束時間 $O(mN \log N)$ 、期待保持時間 $\Omega(Ne^{2N})$ 、メモリ使用量 $O(\Delta \log N)$ ビットとなる決定的プロトコルを提案する。

1 はじめに

個体群プロトコルモデルは、Angluin ら [AAD⁺06] によって導入された計算モデルであり、分散コンピューティングの分野で広く知られている。このモデルは個体と呼ばれる n 個の状態機械から成り、個体はネットワークを成す。このネットワークを個体群と呼ぶ。また、このモデルは、パッシブモバイルセンサネットワークや、化学反応システム、分子計算などに適用可能である。個体は識別子を持たないために互いに区別ができない。個体の状態は、交流と呼ばれる個体間の通信によって更新される。個体間で交流できるか否かは、交流グラフによって決定される。交流グラフは単純有向弱連結グラフ $G = (V, E)$ で表される。個体 u, v が交流可能か否かは (u, v) が E に含まれるか否かで決まり、このとき、 u は送信者、 v は受信者である。このモデルでは、各時刻において1組の個体が交流する。その交流は一様ランダムなスケジューラにより決定される。つまり、 $1/|E|$ の確率で順序対 $(u, v) \in E$ が選ばれ、送信者 u が受信者 v と交流する。

リーダ選挙問題は個体群プロトコルで最も研究されている問題の1つである。この問題は、 n 個の個体の中からただ1つのリーダ個体を選出し、リーダが1つである状況を永遠に保ち続ける問題である。Angluin ら [AAD⁺06] が最初に交流グラフが完全グラフであり、かつ任意の個体の初期状態が指定されている仮定の下で研究した。交流グラフが完全グラフあるとは、任意の個体対が交流可能であることであり、個体の初期状態が指定されているとは、実行開始前の全個体の状態が指定された共通の状態から実行が始まるということである。完全グラフかつ初期状態指定の下では、多数の研究 [AAD⁺06, AG15, BEF⁺21, DS18, AAE08, BGK20, GS18, GSU19, AAG18, MST22, SOI⁺20, GS20] が行われ、時間最適かつ空間最適なプロトコル

*Corresponding Author: kanaya.haruki.kk3@naist.ac.jp

[BGK20] が既に提案されている。個体群プロトコルの時間計算量とは、目的の状況に収束するまでの交流回数の期待値である。交流グラフが任意である場合でもいくつかの研究 [AGR22, ARV22] があり、現在はほぼ時間最適なプロトコル [ARV22] が既に提案されている。個体群にユニークなリーダーがいるとき、個体群プロトコルの問題は高速に解けることが Angluin ら [AAE06] によって証明されているため、リーダー選挙問題を解くことは重要である。

自己安定リーダー選挙問題は、初期状況が**任意**である場合において、リーダー個体を1つ選出し、リーダー個体が1つである状況を永遠に保つ問題である。この問題は、Angluin ら [AAFJ08] が任意グラフでは解けないことを証明した。また、完全グラフでも解けないことが Cai ら [CIW12] によって証明されている。完全グラフや任意グラフ以外では、リンググラフ [YSM21, YSOM23] や k -正則グラフ [CC20] でプロトコルの提案が行われている。完全グラフや任意グラフを研究する場合、多くの研究者は大きくわけて3つの方法で、この問題を研究した。1つ目は、個体が初期知識として正確な個体数 n を知っているという仮定を置く方法である。これは、完全グラフでは [CIW12, BCC⁺21] が研究し、任意グラフでは [SSN⁺21] が研究した。2つ目は、すべての個体にリーダーが存在するか否かを伝えるオラクル $\Omega?$ というものを導入する方法である。これは、完全グラフでは [FJ06]、任意グラフでは [BBB13, CGPB07] が研究した。3つ目は、リーダー個体を1つ選んだあと、リーダー個体が1つあるという状況を永遠に保つという制約を緩める方法である。これは緩安定リーダー選挙問題と呼ばれる。緩安定リーダー選挙問題は、初期状況が**任意**である場合において、比較的短い時間でリーダー個体を1つ選出し、比較的長い時間リーダー個体が1つである状況を保つ問題である。自己安定リーダー選挙と異なり、正確な個体数の知識を必要とせずに、個体数の上界を与えることで緩安定リーダー選挙問題は解くことができる。緩安定リーダー選挙問題は、Sudo ら [SNY⁺12] が完全グラフで最初に研究した。その後、完全グラフ上において、いくつかの研究 [Izu15, SOK⁺20, SEIM21] が提案された。また、任意グラフ上においても以下のようにいくつかの研究がある (表1を参照)。

まず、Sudo ら [SOKM14] は2つのプロトコルを提案した。1つ目は、初期知識として正確な個体数 n の上界 N と、個体の最大次数の上界 Δ を与え、個体が互いに異なる識別子を持っている場合に、期待収束時間 $O(m\Delta N \log n)$ 、期待保持時間 $\Omega(Ne^N)$ 、メモリ使用量 $O(\log N)$ ビットのプロトコルである。2つ目は、初期知識として正確な個体数 n の上界 N と、個体の最大次数の上界 Δ を与え、状態遷移がランダムである場合¹に、期待収束時間 $O(m\Delta^2 N^3 \log N)$ 、期待保持時間 $\Omega(Ne^N)$ 、メモリ使用量 $O(\log N)$ ビットのプロトコルである。ここでいう個体 u の次数とは、個体 u が送信者として交流しうる個体の数と、個体 u が受信者として交流しうる個体の数の合計である。

次に、Sudo ら [SOK⁺19] が、[SOKM14] の収束時間を大幅に削減した。緩安定リーダー選挙プロトコルでは、主にリーダーがいるかどうかを count down timer を用いて判定する。count down timer の仕組みは次の通りである。i) リーダーはタイマーを最大値にセットし続ける。ii) 交流した非リーダーはそれぞれタイマーを1減らす。iii) 非リーダーは自身より高いタイマーを持つ個体と交流した際に、自身のタイマーの値を交流した個体のタイマーの値で更新する。つまり、タイマーの値が a の個体とタイマーの値が b の個体と交流した際に、各々のタイマーの値を $\max(a-1, b-1, 0)$ で更新する。リーダーがいる場合は、タイマーの最大値が十分に大きいとき、任意の個体のタイマーは確率的に0にならない。逆に、リーダーがいない場合は、いずれある個体のタイマーが0になるためリーダーがいないことを確率的に判定できる。Sudo らは Same Speed Timer という特殊なタイマーを導入し、それぞれの個体がタイマーを減らす頻度をほぼ一定にすることで収束時間の削減を成功させた。Same Speed Timer の仕組みは以下の通りである。各個体は直前に交流した個体を覚えておき、連続して同じ個体と交流したとき、かつそのときのみタイマーを減らすことで、全個体のタイマーが減る頻度が一定になる。Sudo らは、初期知識として個体数の上界 N を与え、個体が互いに異なる識別子を持っている場合に、期待収束時間 $O(mN \log n)$ 、期待保持時間 $\Omega(Ne^{2N})$ 、メモリ使用量 $O(\log N)$ ビットのプロトコルを提案した。また、初期知識として個体数の上界 N を与え、状態遷移がランダムである場合に、期待収束時間 $O(mN^2 \log N)$ 、期待保持時間 $\Omega(Ne^{2N})$ 、メモリ使用量 $O(\log N)$ ビットのプロトコルを提案した。さらに、期待保持時間が $\Omega(e^N)$ であるときの期待収束時間の下界が $\Omega(mn)$ であ

¹ 状態遷移がランダムである場合、遷移時に個体は一様乱数を生成し、遷移先を等確率に決定する。

ることを示した。

そして、Sudo ら [SOKM20] が識別子を使用せずに緩安定リーダー選挙を行う決定的プロトコルを提案した。このプロトコルは、初期知識として個体数の上界 N と個体の最大次数の上界 Δ を与えた場合に、期待収束時間 $O(nH_G \log n + mN\Delta^2 \log N)^2$ 、期待保持時間 $\Omega(Ne^{2N})$ 、メモリ使用量 $O(\log N)$ のプロトコルを提案した。ここでいう H_G とは、グラフ上でランダムウォークする 2 つのトークンの hitting time であり、一般に $O(mD)$ である。ただし、 D は G の直径を表す。

表 1: 指数保持時間の緩安定リーダー選挙プロトコルの収束時間と保持時間の一覧。 n は個体数、 N は個体数の上界、 m は交流グラフの辺の数、 D は交流グラフの直径、 Δ は個体の最大次数の上界を表す。すべてのプロトコルは初期知識として個体数の上界 N が与えられる。* が付いているプロトコルは N に加えて個体の最大次数の上界 Δ も与えられる。

| | グラフ | 収束時間 | 保持時間 | メモリ使用量 | 追加の仮定 |
|---------------------------|-----|-------------------------------------|-------------------|--------------------|-------|
| [SNY ⁺ 12] | 完全 | $O(nN \log n)$ | $\Omega(Ne^N)$ | $O(\log N)$ | - |
| [Izu15] | 完全 | $O(nN)$ | $\Omega(e^N)$ | $O(\log N)$ | - |
| [Izu15](下界) | 完全 | $\Omega(nN)$ | $\Omega(e^N)$ | - | - |
| [Izu15](下界) | 完全 | - | $\Omega(e^N)$ | $\Omega(\log N)$ | - |
| [SOKM14]* | 任意 | $O(m\Delta N \log n)$ | $\Omega(Ne^N)$ | $O(\log N)$ | 識別子 |
| [SOKM14]* | 任意 | $O(m\Delta^2 N^3 \log N)$ | $\Omega(Ne^N)$ | $O(\log N)$ | 乱数生成 |
| [SOK ⁺ 19] | 任意 | $O(mN \log n)$ | $\Omega(Ne^{2N})$ | $O(\log N)$ | 識別子 |
| [SOK ⁺ 19] | 任意 | $O(mN^2 \log N)$ | $\Omega(Ne^{2N})$ | $O(\log N)$ | 乱数生成 |
| [SOK ⁺ 19](下界) | 任意 | $\Omega(mN)$ | $\Omega(e^N)$ | - | - |
| [SOKM20]* | 任意 | $O(mnD \log n + mN\Delta^2 \log N)$ | $\Omega(Ne^N)$ | $O(\log N)$ | - |
| 本研究 * | 任意 | $O(mN \log n)$ | $\Omega(Ne^{2N})$ | $O(\Delta \log N)$ | 乱数生成 |
| 本研究 * | 任意 | $O(mN \log N)$ | $\Omega(Ne^{2N})$ | $O(\Delta \log N)$ | - |

1.1 研究成果

本論文では、匿名任意グラフ上で収束時間がほぼ時間最適な緩安定リーダー選挙プロトコル \mathcal{P}_{BC} を提案する。プロトコル \mathcal{P}_{BC} は、初期知識として個体数の上界 N と個体の最大次数の上界 Δ を与え、状態遷移がランダムである場合は、収束時間 $O(mN \log n)$ 、保持時間 $\Omega(Ne^{2N})$ 、メモリ使用量 $O(\Delta \log N)$ ビットとなり、状態遷移が決定的である場合は、収束時間 $O(m(N \log n + \Delta \log N))$ 、保持時間 $\Omega(Ne^{2N})$ 、メモリ使用量 $O(\Delta \log N)$ ビットとなる。

収束時間がほぼ時間最適なプロトコルとして、Sudo ら [SOK⁺19] の \mathcal{P}_{ID2} があるが、これは個体が互いに異なる識別子を持つという強い仮定³を用いている。そのため、 \mathcal{P}_{ID2} よりも弱い仮定のもとで同様の収束時間で収束するプロトコルである。

識別子を使用せずに、状態遷移がランダムである先行研究と比べると、メモリ使用量は $O(\Delta)$ 倍になるが、収束時間は大幅に削減される。 \mathcal{P}_{RD2} [SOK⁺19] の収束時間は $O(mN^2 \log N)$ であり、 \mathcal{P}_{BC} は \mathcal{P}_{RD2} の $1/N$ の収束時間で収束する。

識別子と状態遷移が決定的である先行研究と比べると、メモリ使用量が $O(\Delta)$ 倍になるが、収束時間は大幅に削減される。 \mathcal{P}_{AR} [SOKM20] の収束時間は $O(mnD \log n + mN\Delta^2 \log N)$ である。比較するために、 $D = O(n)$ であり、 $\Delta = O(n)$ 、 $N = \Theta(n) \geq n$ とすると、 $O(mnD \log n + mN\Delta^2 \log N) = O(mn^3 \log n)$ 、状態遷移が決定的な \mathcal{P}_{BC} の収束時間は $O(mN \log N) = O(mn \log n)$ であるため、収束時間は $1/n^2$ にな

²[SOKM20] では $O(mn^2 D \log n + mN\Delta^2 \log N)$ であったが、その後の解析 [SSN⁺21] により $O(mnD \log n + mN\Delta^2 \log N)$ になった。

³個体は低性能なデバイスであるので、一般に識別子や乱数生成器を持つという仮定は強い仮定と言われる。

る。よって既存の仮定よりも弱い仮定で、既存のすべての緩安定プロトコルよりも P_{BC} は速い時間で収束する。また、自己安定リーダ選挙プロトコル P_{rank} [SSN+21] の収束時間は $O(mn^2 D \log n) = O(mn^3 \log n)$ であり、初期知識 n が必要である。 P_{BC} は正確な個体数を必要とせず、 $1/n^2$ の時間で収束し、指数時間リーダが1つである状況を (確率的に) 維持できる。

プロトコル P_{BC} の収束時間を達成するために、Sudo ら [SOK⁺19] と同様に Same Speed Timer を用いる。Same Speed Timer は隣接する個体を区別できる必要があるため、自己安定 2-hop coloring を行う。自己安定 2-hop coloring は Angluin ら [AAFJ08] が最初に研究し、その後 Sudo ら [SOK⁺19] が研究した (表 2 参照)。Sudo ら [SOK⁺19] が提案した 2-hop coloring プロトコルの収束時間は $O(mn\delta \log n)$ であり、かつ状態遷移がランダムであるため、本論文では、新たに収束時間 $O(mn)$ 、メモリ使用量 $O(\Delta \log N)$ ビットの自己安定 2-hop coloring 乱択プロトコル P_{LRU} と、収束時間 $O(m(n + \Delta \log N))$ 、メモリ使用量 $O(\Delta \log N)$ ビットの自己安定 2-hop coloring 決定的プロトコル P'_{LRU} を提案する。 P'_{LRU} では、個体が交流の非対称性を用いて乱数を生成する。個体間で独立な一様乱数が生成できるように、normal coloring を行う、自己安定 normal coloring 決定的プロトコル P_{NC} を提案する。このプロトコルの収束時間は $O(mn \log n)$ 、メモリ使用量は $O(\log N)$ ビットである。

表 2: 任意グラフ自己安定 2-hop coloring プロトコルの収束時間

| | 収束時間 | メモリ使用量 | 初期知識 | 追加の仮定 |
|-----------------------|--------------------------|--------------------|-------------|-------|
| [AAFJ08] | - | $O(\Delta^2)$ | Δ | 乱数生成 |
| [AAFJ08] | - | $O(\Delta^2)$ | Δ | - |
| [SOK ⁺ 19] | $O(mn\delta \log n)$ | $O(\log N)$ | N | 乱数生成 |
| P_{LRU} (本研究) | $O(mn)$ | $O(\Delta \log N)$ | N, Δ | 乱数生成 |
| P'_{LRU} (本研究) | $O(mn + m\Delta \log N)$ | $O(\Delta \log N)$ | N, Δ | - |

2 諸定義

本論文では、1 以上の自然数の集合を \mathbb{N} と表す。

個体群は、単純有向弱連結グラフ $G = (V, E)$ からなる。 V ($|V| \geq 2$) は個体の集合を表し、 $E \subseteq \{(u, v) \in V \times V \mid u \neq v\}$ は個体対の集合であり、個体間が交流できるか否かを表す。 $(u, v) \in E$ であるとき、かつ、そのときのみ送信者 u は受信者 v と交流できる。 G が無向グラフであるとは、 $(u, v) \in E \Leftrightarrow (v, u) \in E$ であることをいう。本論文では、 G は無向グラフであると仮定する。また、 $n = |V|$, $m = |E|$, d を G の直径とし、個体 u の次数を $\delta_u = |\{v \in V \mid (u, v) \in E \vee (v, u) \in E\}|$ 、次数の最大値 $\delta = \max_{u \in V} \delta_u$ と定義する。個体数の上界 N は $N \geq n$ を満たし、個体の最大次数の上界 Δ は $\delta \leq \Delta \leq 2N$ を満たす⁴。

プロトコル \mathcal{P} は 5 項組 (Q, Y, R, T, O) から成り、 Q は個体の状態の有限集合、 Y は出力記号の有限集合、 $R \subset \mathbb{N}$ は乱数の範囲、 $T : Q \times Q \times R \rightarrow Q \times Q$ は状態遷移関数、 $O : Q \rightarrow Y$ は出力関数である。状態 $p \in Q$ の送信者 u が状態 $q \in Q$ の受信者 v と交流したとき、 u, v はそれぞれの状態とこの交流で生成した $r \in R$ に基づいて、それぞれ $(p', q') = T(p, q, r)$ を満たす p', q' に状態を更新する。また、状態 $p \in Q$ の個体は $O(p) \in Y$ を出力する。任意の乱数 $r, r' \in R$ 、任意の状態 $p, q \in Q$ について、 $T(p, q, r) = T(p, q, r')$ が成り立つとき、かつそのときのみ、プロトコル \mathcal{P} は決定的であるという。プロトコル \mathcal{P} が乱択であるとは、 \mathcal{P} が決定的でないことをいう。プロトコル \mathcal{P} のメモリ使用量は、 $\lceil \log |Q| \rceil$ ビットで定義される。

すべての個体の状態を表す状況は $C : V \rightarrow Q$ と定義される。プロトコル \mathcal{P} によるすべての状況の集合を $\mathcal{C}_{all}(\mathcal{P})$ とする。状況 C が交流 $e = (u, v)$ と乱数 $r \in R$ によって C' に遷移するとは、 $(C'(u), C'(v)) = T(C(u), C(v), r)$ かつ $\forall w \in V - \{u, v\} : C'(w) = C(w)$ が成り立つことをいう。状況 C が交流 e と乱数 r によって C' に遷移することを $C \xrightarrow{e, r} C'$ と表す。一様ランダムスケジューラ $\Gamma = \Gamma_0, \Gamma_1, \dots$ は各時刻でど

⁴ 初級知識として個体数の上界 N のみが与えられた場合、個体の最大次数は高々 $2N$ 以下である。よって、 $\Delta \leq 2N$ が成り立つ。

の個体対が交流するかを決める。ただし時刻 $t \geq 0$ において、 $\Gamma_t \in E$ であり、任意の $(u, v) \in E$ について、 $\Pr(\Gamma_t = (u, v)) = 1/m$ が成り立つ。また、乱数の無限列 $\Lambda = R_0, R_1, \dots$ は各時刻で生成される乱数を表す。ただし、 $R_t \in R$ は一様乱数であり、任意の $r \in R$ について、 $\Pr(R_t = r) = 1/|R|$ を満たす。初期状態 C_0 とスケジューラ Γ 、乱数列 Λ を与えたプロトコル \mathcal{P} の実行は $\Xi_{\mathcal{P}}(C_0, \Gamma, \Lambda) = C_0, C_1, \dots$ と定義される。ただし、 $t \geq 0$ について、 $C_t \xrightarrow{\Gamma_t, R_t} C_{t+1}$ が成り立つ。文脈上 Γ や Λ が明らかである場合、単に $\Xi_{\mathcal{P}}(C_0)$ と表記する、状況集合 S が安全であるとは、 $C_0 \in S$ から始まる任意の実行 C_0, C_1, \dots について、 $C_i \notin S$ を満たすような $i \in \mathbb{N}$ が存在しないことをいう。プロトコルがサイレントであるとは、安全状況に達したあと、すべての個体の状態が変化しないことをいう。

個体群プロトコルの問題を解くプロトコル \mathcal{P} について、期待保持時間と期待収束時間は次のように定義される。問題の仕様を \mathcal{SC} とする。仕様とは、状況の系列であり、問題が満たすべき条件を示す。任意の $C \in \mathcal{C}_{all}(\mathcal{P})$ 、任意の一様ランダムスケジューラ Γ 、任意の乱数の無限列 Λ について、実行 $\Xi_{\mathcal{P}}(C, \Gamma, \Lambda)$ が \mathcal{SC} を満たし続ける交流回数の期待値を期待保持時間といい、 $\text{EHT}_{\mathcal{P}}(C, \mathcal{SC})$ と定義する。つまり、期待保持時間は C_0 から始まる実行 $C_0, C_1, \dots, C_x, \dots$ において、 C_0 から C_x まで \mathcal{SC} を満たすような最大値 x の期待値である。任意の状況集合 $S \subseteq \mathcal{C}_{all}(\mathcal{P})$ 、任意の状況 $C \in \mathcal{C}_{all}(\mathcal{P})$ 、任意の一様ランダムスケジューラ Γ 、任意の乱数の無限列 Λ について、実行 $\Xi_{\mathcal{P}}(C, \Gamma, \Lambda)$ が開始してから S に属する状況に遷移するまでの交流回数の期待値を期待収束時間といい、 $\text{ECT}_{\mathcal{P}}(C, S)$ と定義する。ある計算が高確率で終了するとは、確率 $1 - O(n^{-c}) (c \geq 1)$ で終了することをいう。

リーダー選挙問題は全個体が L または F を出力する。ここで、 L はリーダー、 F は非リーダーを表す。リーダー選挙の仕様は \mathcal{LE} で定義される。実行 $\Xi_{\mathcal{P}}(C_0) = C_0, C_1, \dots, C_x, \dots$ において、 $O(C_0(u)) = L$ であるような個体 u が存在し、 $\forall i \in [1, x] : O(C_i(u)) = L$ かつ $\forall i \in [0, x], \forall v \in V - \{u\} : O(C_i(v)) = F$ を満たすとき、 C_0, \dots, C_x は \mathcal{LE} を満たすという。

定義 1 (緩安定リーダー選挙 [SNY⁺12])。プロトコル \mathcal{P} が (α, β) -緩安定リーダー選挙プロトコルであるとは、 $\max_{C \in \mathcal{C}_{all}(\mathcal{P})} \text{ECT}_{\mathcal{P}}(C, S) \leq \alpha$ かつ $\min_{C \in S} \text{EHT}_{\mathcal{P}}(C, \mathcal{LE}) \geq \beta$ を満たすような状況の集合 $S \subseteq \mathcal{C}_{all}(\mathcal{P})$ が存在することをいう。

プロトコル \mathcal{P} が自己安定であるとは、問題の仕様を満たす安全な状況集合が存在し、かつ $\mathcal{C}_{all}(\mathcal{P})$ に属する任意の状況からその安全状況に達することが可能であることをいう。

定義 2。プロトコル \mathcal{P} が自己安定 normal coloring プロトコルであるとは、任意の初期状況 $C_0 \in \mathcal{C}_{all}(\mathcal{P})$ 、任意の実行 $\Xi_{\mathcal{P}}(C_0) = C_0, C_1, \dots, C_x, \dots$ において、 $i = 1, 2, 3, \dots$ について $\forall v \in V : O(C_x(v)) = O(C_{x+i}(v))$ かつ $\forall v, \forall u \in V : (u, v) \in E \Rightarrow O(C_x(u)) \neq O(C_x(v))$ が成り立つような非負整数 x が存在することをいう。

定義 3。プロトコル \mathcal{P} が自己安定 2-hop coloring プロトコルであるとは、任意の初期状況 $C_0 \in \mathcal{C}_{all}(\mathcal{P})$ 、任意の実行 $\Xi_{\mathcal{P}}(C_0) = C_0, C_1, \dots, C_x, \dots$ において、 $i = 1, 2, 3, \dots$ について $\forall v \in V : O(C_x(v)) = O(C_{x+i}(v))$ かつ $\forall v, \forall u, \forall w \in V : (u, v) \in E \wedge (v, w) \in E \Rightarrow O(C_x(u)) \neq O(C_x(w))$ が成り立つような非負整数 x が存在することをいう。

3 主定理

本論文では主定理を述べるにとどめ、詳細は当日の発表で紹介する。

定理 4。個体数の上界 N と個体の最大次数の上界 Δ を与えたとき、 \mathcal{P}_{LRU} は自己安定 2-hop coloring 乱択プロトコルであり、その収束時間は高確率、期待値ともに $O(mn)$ である。

定理 5。個体数の上界 N を与えたとき、 \mathcal{P}_{NC} はサイレントな自己安定 normal coloring 決定的プロトコルであり、その収束時間は高確率、期待値ともに $O(mn \log n)$ である。

定理 6. 個体数の上界 N と個体の最大次数の上界 Δ を与えたとき、 P'_{LRU} は自己安定 2-hop coloring 決定的プロトコルであり、その収束時間は高確率、期待値ともに $O(m(n + \Delta \log N))$ である。

$\tau \geq \max(2d, \lceil \log N \rceil / 2, 15 + 3 \log n)$ とする。

定理 7. 個体数の上界 N と個体の最大次数の上界 Δ を与えたとき、 \mathcal{P}_{BC} は任意グラフ上での $(O(m(n + \tau \log n)), \Omega(\tau e^\tau))$ -緩安定リーダ選挙乱択プロトコルである。

定理 8. 個体数の上界 N と個体の最大次数の上界 Δ を与えたとき、 \mathcal{P}_{BC} は任意グラフ上での $(O(m(n + \Delta \log N + \tau \log n)), \Omega(\tau e^\tau))$ -緩安定リーダ選挙決定的プロトコルである。

参考文献

- [AAD⁺06] Dana Angluin, James Aspnes, Zoë Diamadi, Michael J. Fischer, and René Peralta. Computation in networks of passively mobile finite-state sensors. *Distributed Computing*, 18(4):235–253, 2006.
- [AAE06] Dana Angluin, James Aspnes, and David Eisenstat. Fast computation by population protocols with a leader. In *Distributed Computing*, pages 61–75, 2006.
- [AAE08] Dana Angluin, James Aspnes, and David Eisenstat. Fast computation by population protocols with a leader. *Distributed Computing*, 21(3):183–199, 2008.
- [AAFJ08] Dana Angluin, James Aspnes, Michael J. Fischer, and Hong Jiang. Self-stabilizing population protocols. *ACM Trans. Auton. Adapt. Syst.*, 3(4), 2008.
- [AAG18] Dan Alistarh, James Aspnes, and Rati Gelashvili. Space-optimal majority in population protocols. In *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms*, page 2221–2239, 2018.
- [AG15] Dan Alistarh and Rati Gelashvili. Polylogarithmic-time leader election in population protocols. In *Automata, Languages, and Programming*, pages 479–491, 2015.
- [AGR22] Dan Alistarh, Rati Gelashvili, and Joel Rybicki. Fast Graphical Population Protocols. In *25th International Conference on Principles of Distributed Systems (OPODIS 2021)*, volume 217, pages 14:1–14:18, 2022.
- [ARV22] Dan Alistarh, Joel Rybicki, and Sasha Voitovich. Near-optimal leader election in population protocols on graphs. In *Proceedings of the 2022 ACM Symposium on Principles of Distributed Computing*, page 246–256, 2022.
- [BBB13] Joffroy Beauquier, Peva Blanchard, and Janna Burman. Self-stabilizing leader election in population protocols over arbitrary communication graphs. In *Principles of Distributed Systems*, pages 38–52, 2013.
- [BCC⁺21] Janna Burman, Ho-Lin Chen, Hsueh-Ping Chen, David Doty, Thomas Nowak, Eric Severson, and Chuan Xu. Time-optimal self-stabilizing leader election in population protocols. In *Proceedings of the 2021 ACM Symposium on Principles of Distributed Computing*, page 33–44, 2021.

- [BEF⁺21] Petra Berenbrink, Robert Elsässer, Tom Friedetzky, Dominik Kaaser, Peter Kling, and Tomasz Radzik. Time-space trade-offs in population protocols for the majority problem. *Distributed Computing*, 34(2):91–111, 2021.
- [BGK20] Petra Berenbrink, George Giakkoupis, and Peter Kling. Optimal time and space leader election in population protocols. In *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing*, page 119–129, 2020.
- [CC20] Hsueh-Ping Chen and Ho-Lin Chen. Self-stabilizing leader election in regular graphs. In *Proceedings of the 39th Symposium on Principles of Distributed Computing*, page 210–217, 2020.
- [CGPB07] Davide Canepa and Maria Gradinariu Potop-Butucaru. Stabilizing leader election in population protocols. Research Report RR-6269, INRIA, 2007.
- [CIW12] Shukai Cai, Taisuke Izumi, and Koichi Wada. How to prove impossibility under global fairness: On space complexity of self-stabilizing leader election on a population protocol model. *Theory of Computing Systems*, 50(3):433–445, 2012.
- [DS18] David Doty and David Soloveichik. Stable leader election in population protocols requires linear time. *Distributed Computing*, 31(4):257–271, 2018.
- [FJ06] Michael Fischer and Hong Jiang. Self-stabilizing leader election in networks of finite-state anonymous agents. In *Principles of Distributed Systems*, pages 395–409, 2006.
- [GS18] Leszek Gąsieniec and Grzegorz Stachowiak. Fast space optimal leader election in population protocols. In *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms*, page 265–266, 2018.
- [GS20] Leszek Gąsieniec and Grzegorz Stachowiak. Enhanced phase clocks, population protocols, and fast space optimal leader election. *J. ACM*, 68(1), 2020.
- [GSU19] Leszek Gąsieniec, Grzegorz Stachowiak, and Przemysław Uznanski. Almost logarithmic-time space optimal leader election in population protocols. In *The 31st ACM Symposium on Parallelism in Algorithms and Architectures*, page 93–102, 2019.
- [Izu15] Taisuke Izumi. On space and time complexity of loosely-stabilizing leader election. In *Structural Information and Communication Complexity*, pages 299–312, 2015.
- [MST22] Othon Michail, Paul G. Spirakis, and Michail Theofilatos. Simple and fast approximate counting and leader election in populations. *Information and Computation*, 285(A):104698, 2022.
- [SEIM21] Yuichi Sudo, Ryota Eguchi, Taisuke Izumi, and Toshimitsu Masuzawa. Time-Optimal Loosely-Stabilizing Leader Election in Population Protocols. In *35th International Symposium on Distributed Computing (DISC 2021)*, volume 209, pages 40:1–40:17, 2021.
- [SNY⁺12] Yuichi Sudo, Junya Nakamura, Yukiko Yamauchi, Fukuhito Ooshita, Hirotsugu Kakugawa, and Toshimitsu Masuzawa. Loosely-stabilizing leader election in a population protocol model. *Theoretical Computer Science*, 444:100–112, 2012.

- [SOI⁺20] Yuichi Sudo, Fukuhito Ooshita, Taisuke Izumi, Hirotsugu Kakugawa, and Toshimitsu Masuzawa. Time-optimal leader election in population protocols. *IEEE Transactions on Parallel and Distributed Systems*, 31(11):2620–2632, 2020.
- [SOK⁺19] Yuichi Sudo, Fukuhito Ooshita, Hirotsugu Kakugawa, Toshimitsu Masuzawa, Ajoy K. Datta, and Lawrence L. Larmore. Loosely-stabilizing leader election for arbitrary graphs in population protocol model. *IEEE Transactions on Parallel and Distributed Systems*, 30(6), 2019.
- [SOK⁺20] Yuichi Sudo, Fukuhito Ooshita, Hirotsugu Kakugawa, Toshimitsu Masuzawa, Ajoy K. Datta, and Lawrence L. Larmore. Loosely-stabilizing leader election with polylogarithmic convergence time. *Theoretical Computer Science*, 806:617–631, 2020.
- [SOKM14] Yuichi Sudo, Fukuhito Ooshita, Hirotsugu Kakugawa, and Toshimitsu Masuzawa. Loosely-stabilizing leader election on arbitrary graphs in population protocols. In *Principles of Distributed Systems*, pages 339–354, 2014.
- [SOKM20] Yuichi Sudo, Fukuhito Ooshita, Hirotsugu Kakugawa, and Toshimitsu Masuzawa. Loosely stabilizing leader election on arbitrary graphs in population protocols without identifiers or random numbers. *IEICE Transactions on Information and Systems*, E103.D(3):489–499, 2020.
- [SSN⁺21] Yuichi Sudo, Masahiro Shibata, Junya Nakamura, Yonghwan Kim, and Toshimitsu Masuzawa. Self-stabilizing population protocols with global knowledge. *IEEE Transactions on Parallel and Distributed Systems*, 32(12):3011–3023, 2021.
- [YSM21] Daisuke YOKOTA, Yuichi SUDO, and Toshimitsu MASUZAWA. Time-optimal self-stabilizing leader election on rings in population protocols. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, E104.A(12):1675–1684, 2021.
- [YSOM23] Daisuke Yokota, Yuichi Sudo, Fukuhito Ooshita, and Toshimitsu Masuzawa. A near time-optimal population protocol for self-stabilizing leader election on rings with a poly-logarithmic number of states. In *Proceedings of the 2023 ACM Symposium on Principles of Distributed Computing*, page 2–12, 2023.