Vol.2018-AL-170 No.13
2018/11/13

準線形時間ランデブーの可能性について

江口 僚太[1,a)]　北村 直暉[1,b)]　泉 泰介[1,c)]

概要：ランデブー問題は分散計算分野において広く研究されている問題である．ランデブー問題の基本的な設定は，グラフ $G = (V, E)$ 上に配置された 2 つのエージェントを，最終的に同一な頂点上に移動し停止させるというものである．本研究では，エージェントの動作がラウンドに基づき同期的に動く仮定のもとで，ランデブー達成に必要はラウンド数をできるだけ少なくすることである．特に，G の頂点数 n に対する準線形時間でランデブーが可能かどうかを探求する．最小次数が比較的大きいグラフにおいて，距離 1 に配置された 2 エージェントが高確率で準線形時間でランデブーを達成することを示す．より具体的には，最小次数 δ に関して $\delta = \Omega(n)$ となる任意のグラフ G において，距離 1 に配置されたエージェントが $O(\sqrt{n}\log n)$ 時間で高確率でランデブーを完了するアルゴリズムを提案する．

On the Possibility of Sublinear-Time Rendezvous

Eguchi Ryota[1,a)]　Kitamura Naoki[1,b)]　Izumi Taisuke[1,c)]

1. はじめに

ランデブー問題は分散計算分野において広く研究されている問題である．ランデブー問題の基本的な設定は，グラフ $G = (V, E)$ 上に配置された 2 つのエージェントを，最終的に同一な頂点上に移動し停止させるというものであり，モバイルエージェントおよび自律分散ロボット群のための分散アルゴリズム設計における重要な基本問題の一つとして認識されている．ランデブー問題の本質的な困難さの一つとして，対称性の破壊と呼ばれる概念を挙げることができる．例えば，4 頂点のリングネットワークにおいて，隣合わない 2 頂点上に 2 台のエージェントが配置されている状況を考える[*1]．このような状況で 2 台のエージェントが同一の決定性アルゴリズムで動作する場合，2 台のエージェントは対称に動作するため，その相対位置が距離 2(互いに隣接していない) である状況を永遠に脱出することができない．すなわち，いかなるアルゴリズムをもってしてもこの状況ではランデブーを達成することができない．このことから分かるように，一般にランデブー問題が可解であるためには，2 台のエージェントの動作を非対称にするための何らかの能力を系が備えている必要がある．ランデブー問題を扱う多くの既存研究において，どのような仮定あるいはモデルがそのような能力を与えるかという点がこれまでに検討されている ．本研究では，上述の研究とは異なり，対称性の破壊を必要としないモデル（乱択および 2 エージェントが異なるアルゴリズムを実行することを許す）を仮定し，その上でランデブーを達成するために必要な時間計算量に注目する．2 エージェントで異なる動作を許す場合，ランデブー問題はグラフ探索問題を用いて容易に解くことができる．具体的には，一方のエージェントが初期ノードに停止し，もう一方のエージェントがグラフを探索することでランデブー問題を解くことができる．すなわち，計算量の観点からは，グラフ探索を実現可能な時間計算量がランデブー問題の自明な上界になる．一方で下界に関しては，入力として与えられたグラフ G における 2 エージェントの初期配置における距離の半分が自明な下界となる．最悪時 (例えば n 頂点リングネットワーク) においてこの下界は明らかに $\Omega(n)$ ラウンドとなる．一方で，初期位置の距離が n よりも大幅に小さいときは，グラフ探

[1] 名古屋工業大学
[a)] 30514002@stn.nitech.ac.jp
[b)] 29414045@stn.nitech.ac.jp
[c)] t-izumi@nitech.ac.jp

[*1] 厳密には，各頂点上で接続されている辺に対して割り当てられているポート番号も対称に（例えば時計回り順方向は 1，逆方向は 0 のように) に割り当てられているとする．

ⓒ 2018 Information Processing Society of Japan

索に基づくようなアプローチは全頂点を網羅的に探索するという性質上 $\Omega(n)$ ラウンドより高速にランデブーすることはできないため，必ずしも最適な戦略とはいえない．理想的には，両エージェントが近接している場合，グラフ全体の探索を用いず準線形時間でランデブーを完了するアルゴリズムが望ましい．この要求を満たす解法が存在するかどうかは，乱択や ID，十分なメモリ領域が利用できると仮定しても明らかではない．準線形時間のランデブーを直接的に目的とした研究は著者らの知る限り知られていないが，Collins らによる研究 [10]，Das らによる研究 [15]，および Anderson らによる研究 の研究においてある種のモデル上でそれが達成可能であることが間接的に示されている．Collins らの研究においては，2 エージェントがグラフ全体の知識および自身の初期位置を知識として持つ状況において，距離 d 離れたエージェントのランデブーを $O(d\log^2 n)$ 時間で決定的にランデブーを完了するほぼ最適なアルゴリズムが提案されている．Das らは，エージェントが各ラウンドで相手との距離を検知することができるというモデルを仮定し，その上で $O(\Delta(d+\log l))$ 時間でランデブーを完了するアルゴリズムを与えた．ここで，l は両エージェントの ID の最小値（$l=\min(l_1,l_2)$）であり，Δ はグラフの最大次数である．また下界として同様のモデルにおいてランデブーを達成するためには $\Omega(\Delta(d+\log l/\log\Delta))$ ラウンドを必要とすることを示した． Anderson らの結果では，完全グラフ上での 2 エージェントのランデブーが期待時間 $O(\sqrt{n})$ で達成できることが示されている．この研究において示されている手法は頂点集合のランダムサンプリングに基づいており，エージェントのランダムウォークに基づくサンプリングを組み合わせると，一般のグラフ G に対する $O(\tau(G)\sqrt{n})$ 期待時間のランデブーを達成することが可能である (ここで，$\tau(G)$ は入力グラフ G 上のランダムウォークの混合時間を表す)．エキスパンダーグラフのような高いコンダクタンス値を持つグラフでは $\tau(G)$ が $O(\log n)$ 程度で抑えられるが，例えば橋を持つようなグラフ（小さいカットを持つグラフ）では $\tau(G)=\Omega(\sqrt{n})$ となりうるので，準線形時間でのランデブーを達成することができない．

1.1 研究結果

本研究では，我々はエージェントの初期位置が近接している場合において，入力グラフ G のトポロジに関する事前知識を持たない場合における**準線形時間ランデブー**，すなわち $o(n)$ ラウンドでのランデブー可能性について考える．特に 2 エージェントの初期配置距離が 1 の場合における準線形ランデブー達成可能性を考える．計算モデルとしては，次のようなモデルを考える．エージェントは事前にグラフの頂点数や最小次数，最大次数などの部分的な情報を持つものとする．また，ある頂点 v 上にエージェントが存在するときその隣接頂点 $N(v)$ の情報を観察することのできるモデル（1 ホップ視認性）を考える．また，各頂点はエージェントにより情報を書き残すためのスペース（白板）を備えるものとする．このモデルの上で，最小次数 $\delta_G=\Omega(n)$ となるグラフにおいて，距離 1 上の 2 頂点に配置されたエージェントによる準線形時間ランデブーを高確率で達成するアルゴリズムを提案する．具体的には，この前提のもとで，提案アルゴリズムは $O(\sqrt{n}\log n)$ 時間で高確率でランデブーを達成する．

1.2 関連研究

前述の通り，ランデブー問題は，アルゴリズムが決定性かどうか，対象とするグラフクラス，エージェント動作の一様性，システムの同期性など様々な要因によって可解性が変動する．そのため，さまざまなモデル上で研究がおこなわれている．準線形時間のランデブーに関しての既存研究に関しては前節で述べたので，ここではそれ以外の既存研究，特に計算時間やメモリ使用量等の複雑性解析について検討しているものを中心に概観する．ランデブー問題の包括的な解説に関しては教科書 [4], [5] やサーベイ論文 [2], [21], [26] に詳しいので，参照されたい．

ランデブー問題の初期の研究の興味は対称性の破壊の可能性についてであったため，特にリングネットワーク上でのランデブー問題の可解性が各種モデルにおいて広く検討されてきた [16], [19], [22]．この文脈においては計算複雑性に関しての議論は必ずしも進んではいなかった，リングに次いでよく検討されているトポロジのクラスとして木を挙げることができる．木上のランデブーに関しては，計算時間ならびにエージェントのメモリ使用量に関しての複雑性に関しての議論が進んでいる．Baba ら [7] は，エージェントが $O(n)$ ビットのメモリを備えているという仮定の下で，線形時間 (すなわち $O(n)$ 時間) でランデブーを達成するアルゴリズムを示すとともに，それが必要メモリ量の意味で最適であることを示している．Czyzowicz ら [12] はこの結果を一般化して，k ビットを使用するエージェント $\Theta(n+n^2/k)$ 時間でランデブーするアルゴリズムを示している．木上のランデブーを最適なメモリ使用量 ($O(\log n)$ ビット) で達成するアルゴリズムは Fraigniaud ら [20] により示されている．

一般のグラフにおけるランデブーの可解性については，文献 [8], [11], [13], [17] などで検討されている．特に文献 [11] では，一様なエージェントのランデブーに必要なメモリ量の検討が行われている．結果として，$\Theta(\log n)$ ビットが 2 エージェントの任意の匿名グラフにおいてランデブーを達成するために必要なメモリ量であることを示した．最近では Miller ら [25] が，ランデブーまでに必要な時間（ラウンド数）と，コスト（エージェントが移動する辺の総本数）との間のトレードオフを解明している．またアドバイスと呼ばれる，オラクルからの情報を用いることで高速にラン

デブーを達成する研究 [18], [23], [24] なども知られている．

確率的な動作を許したランデブー問題は，しばしばランダムウォークの理論の一部として検討されている．与えられたグラフ上でランダムウォークする 2 つのトークンが同じ頂点で出会うまでに必要な時間はそのグラフの合流時間 (meeting time) と呼ばれており，いくつかの研究結果が知られている [9], [27]．一般のマルコフ連鎖の解析における応用としてのランデブー問題は制御最適化の文脈において多く研究されている [1], [3], [6], [14], [28], [29]．

2. 準備

2.1 モデルとエージェントの振る舞い

本稿では，n 頂点の無向グラフ $G = (V, E)$ 上で 2 エージェントのランデブーを考える．グラフの各頂点は互いに異なる識別子 $\{v_1, v_2, \ldots, v_n\}$ を持つ．グラフ G の最小次数を δ_G，最大次数を Δ_G で表記する．また，ある頂点 v の隣接頂点集合を $N(v)$ と表記する．すなわち，$N(v) = \{v_i | (v, v_i) \in E\}$ とする．対象としているグラフが明らかな場合，最小次数 δ_G，最大次数 Δ_G の添字は簡単のため省略される場合がある．

グラフ上のある 2 頂点には確率的ランダムアクセス機械としてモデル化される 2 つのエージェント A, B が存在する．その 2 エージェントは互いに異なる ID$\{A, B\}$ を持ち，初期状態から非対称的に異なる動作が可能である．エージェントは内部状態として記憶領域 $M = \{0, 1\}^*$ を持つ．本稿が提案するモデルはエージェントの内部計算において利用される記憶領域および計算能力に関して特に仮定を置かないが，提案するすべてのアルゴリズムにおいて，内部計算は n の多項式領域 (ビット数) を用いて，多項式時間計算で実現することができる．

2 つのエージェントは同一の頂点に存在するとき，互いに相手の存在を検知することができる．また，任意のエージェントはある頂点 v に存在するとき，自分が現在滞在している頂点の識別子を知ることができるとともに，その頂点に隣接している頂点の集合を（その識別子の集合として）知ることができる．ここで，頂点 v の隣接ノードの集合を $N(v) = \{w | (v, w) \in E\}$ と表記する．また，頂点 v を含めたものを $N^+(v) = N(v) \cup \{v\}$ と表記する．各頂点には白板と呼ばれる永続的な記憶領域が備えられており，頂点 v にいるエージェントは v の白板の内容に対して閲覧および書き換えをすることができるものとする．形式的には白板は n 次元ベクトル W^n で表現される．本稿で仮定するモデルは白版のサイズに関して特に制限を置かないが，提案するアルゴリズムはすべて O(1) ビットのサイズの白版を用いて実現することができる．また，一部のアルゴリズムに関しては白版を用いない（すなわち，サイズが 0 ビット）で実現することが可能である．この 2 つのモデルをそれぞれ**白板付きモデル**，**白板なしモデル**として明示的に区別する．

2 つのエージェントは，離散単位時間 $t = 0, 1, 2, \ldots$ に従って同期的に動作する（以降ラウンドと呼ぶ）．時刻 t における状況 C は 5-組 $C \in \mathcal{C} = (V \times M)^2 \times W^n$ である．各ラウンドにおいて，アルゴリズム $\mathcal{A}$ はエージェントのメモリ及び滞在ノードで上でアクセス可能な情報 (現ノードの識別子，隣接ノードの識別子，現ノード上の白版の情報，現ノードにおける他エージェントの存在の有無）に基づき，隣接頂点に移動するか，その頂点にとどまるかを計算する．各ラウンドにおける移動は当該ラウンド中に必ず目的地ノードへと到達できるものとし，辺上に留まっているような移動中の状態は考えないものとする．また，内部計算において，エージェントは現在滞在しているノードの白版の情報を改変することが可能である[*2]．

形式的にはアルゴリズムと実行は次のように定義される．Π^V,Π^W をそれぞれ V, W 上の確率分布すべてからなる集合とする．アルゴリズムはこれらの確率分布を出力とする関数 $\mathcal{A} : M \times \{A, B\} \times V \times 2^V \times W \times \{0, 1\}^* \to \Pi^V \times \Pi^W$ である．入力はそれぞれエージェントのメモリ及び ID，滞在ノード v，ノード v の隣接ノードの集合 $N^+(v)$，ランダムビットに対応している．出力される確率分布 $(\pi^V, \pi^W) \in \Pi^V \times \Pi^W$ はそれぞれエージェントが次に訪問するノードの確率分布，また現在の滞在ノードの白板上に書き残される v に対して $\forall v' \notin N^+(v) \Pr[\pi^V(v')] = 0$ 及び $\forall v' \notin N^+(v), \forall w \in W \Pr[\pi^W(v', w)] = 0$ を満たすものとする．実行は状況 C の無限列 $C_0, C_1, C_2, \ldots$ である．状況 $C_i = (v_i^A, m_i^A, v_i^B, m_i^B, w_1, \ldots, w_n)$ 及び $C_{i+1} = (v_{i+1}^A, m_{i+1}^A, v_{i+1}^B, m_{i+1}^B, w_1', \ldots, w_n')$ に対して次の制約を満たすものとする．$v_{i+1}^A \in N^+(v_i^A)$,$\Pr[\pi(v_{i+1}^A)] > 0$，$v_{i+1}^B \in N^+(v_i^B)$,$\Pr[\pi^V(v_{i+1}^B)] > 0$ であり，$\forall j (j \in V \setminus \{v_i^A, v_i^B\}) w_j = w_j'$ であり $\Pr[\pi^W(v_i^A, w'_{v_i^A})] > 0$ かつ $\Pr[\pi^W(v_i^B, w'_{v_i^A})] > 0$.

2.2 ランデブー問題

ランデブー問題とは，与えられたグラフ G のある 2 頂点に配置されたエージェント同じ頂点に移動し停止する問題として定義される．形式的には，あるアルゴリズムの実行において，時刻 t にエージェント A および B が同一頂点に滞在しているとき，その実行において時刻 t でランデブーが達成されていると呼ぶ[*3]．以降エージェント A, B の初

[*2] モデルを正確に定義するためには，同一ノードに滞在している 2 つ以上のエージェントが同時に同じ白版を書き換えた場合の振る舞いについて規定する必要があるが，本研究で考える 2 エージェントのランデブー問題では，一般性を失うことなく 2 つのエージェントが同一ノードに滞在しているときをアルゴリズムの終了とみなすことができる．そのため，白版への同時並行的な書き込みに関しては考えないこととする．

[*3] 同期システムにおける 2 エージェントのランデブー問題では，一般性を失うことなく，一度出会った 2 エージェントがそれ以降動作を停止する仮定してよい．すなわち，時刻 t にランデブーを達成したエージェントは，それ以降の任意の時刻 $t' > t$ においても

期位置を v_A, v_B と表記する．本研究では特に，エージェントの初期位置に関して制約を置いたランデブー問題を考える．

定義 1 (制限付きランデブー)**.** グラフ $G = (V, E)$ において，頂点対の集合 $I \subseteq V \times V$ がエージェントの可能な初期位置 (v_A, v_B) の集合として与えられたとする．組 (G, I) をグラフ G と制限初期位置 I で記述されるインスタンスと呼ぶ．インスタンス (G, I) に対して，エージェントの初期位置 (v_a, v_b) として I 中の任意の要素を選びアルゴリズム $\mathcal{A}$ を実行することを考える．このとき，$\mathcal{A}$ の実行において確率 p 以上で時刻 t に 2 エージェントがランデブーを達成しているとき，$\mathcal{A}$ はインスタンス (G, I) に対して確率 p および時間 t でランデブー問題を解くと呼ぶ．さらに，$\mathcal{P} = \{(G_0, I_0), (G_1, I_1), \cdots\}$ をインスタンスの集合とし，ある関数 $f : \mathbb{N} \to \mathbb{N}$ が存在し，アルゴリズム $\mathcal{A}$ が $\mathcal{P}$ 中の任意のインスタンス $((V, E), I) \in \mathcal{P}$ に対して確率 p および時間 $f(|V|)$ でランデブーを達成するとき，$\mathcal{A}$ はクラス $\mathcal{P}$ に対して確率 p および実行時間 $f(n)$ でランデブーを達成するという．

本稿では初期位置の制限として，2 エージェント間の距離に関しての上限 d を仮定するケース，すなわち入力グラフ G に対して $I_d^G = \{(v_A, v_B) | \ \mathrm{dist_G}(v_A, v_B) \leq d\}$ を主に考える．最大次数が Δ 以内，最小次数が δ 以上であるようなすべてのグラフの集合を $\mathcal{G}(\Delta, \delta)$ としたとき，インスタンス集合 $\mathcal{P}_d = \{(G, I_d^G) \mid G \in \mathcal{G}(\Delta, \delta)\}$ により制限されたランデブー問題を (Δ, δ, d)-ランデブー問題と呼ぶ．

本稿を通じて，何らかの事象が確率 $1 - 1/n^{\Theta(1)}$ 以上で成立することを単に**高確率**で成功すると言う．次節で提案アルゴリズムは高確率でランデブーを達成する，すなわち $p \geq 1 - 1/n^{\Theta(1)}$ であることが保証される．

3. ランデブーアルゴリズム

3.1 概要

アルゴリズムの記述のために，2 つの頂点集合対 $(X, Y)(X, Y \subseteq V)$ に対する隣接頂点の密集度という性質を定義する．頂点集合 X 内の各頂点の隣接頂点の和を $N(X) = \bigcup_i N(x_i)$ で表す．また，$N^+(X)$ を $N^+(X) = \bigcup_i N^+(x_i)$ と定義する．

定義 2. 頂点集合の対 $X, Y \subseteq V$ とある定数 α とある整数 $\nu (1 \leq \nu \leq n)$ に対して，対 (X, Y) が (α, ν)-密集しているとは，

$$|N(X) \cap N(Y)| \geq \alpha^{-1}\nu$$

が成立するときいう．

なお，X, Y がただ一つの要素 x, y からなる場合集合を表す記号 $\{\}$ を省略する．例えば，対を $(X, y), (x, Y), (x, y)$ のように表記する．また，対 (X, Y) が (α, ν)-密集であるとき，X が Y と (α, ν)-密集するという．

本節を通じて，アルゴリズムはネットワーク全体の最低次数 δ および $\log n$ の値を既知であると仮定している．本稿を通して $\delta = \Omega(n)$ を仮定しているため，実際には v_0^A の次数 $|N(v_0^A)|$ を δ の近似値，またその対数値 $\log |N(v_0^A|$ を $\log n$ の近似値として用いることこの問題は解消できる（厳密には，$\log n = \beta \log |N(v_0^A|$ としたときの β の値はアルゴリズムの成功確率に影響するが，ランデブーが失敗したとき，$\log |N(v_0^A|$ の 2 倍を $\log n$ の近似値にして再度アルゴリズムを実行，それでも失敗した場合は 4 倍の値を近似値として実行，のように反復を繰り返すことで，高確率でアルゴリズムを成功させることが可能である．また，δ の近似値もまた $\Delta/\delta = O(1)$ の誤差が生じうるが，こちらはランデブーが失敗したとき値を 1/2 倍していくことで同様に高確率でアルゴリズムを成功させることが可能である）．よって，以降のアルゴリズムでは δ および $\log n$ の値を断りなく用いる．また以降ある定数 γ により，$\delta = \gamma n$ と表す．$\delta = \Omega(n)$ であることより $\gamma = \Theta(1)$ である．

提案アルゴリズムでは，高速なランデブーが可能になるようにエージェント A は次の性質を持つ $N^+(v_0^A)$ の部分集合 $S \subseteq N^+(v_A)$ を探し出す．

- 任意の $y \in N(v_0^A) \setminus S$ に対して，対 (S, y) が $(32, \delta)$-密集している．つまり，v_0^B は S 内に存在するか，S 外だが (S, v_0^B) が $(32, \delta)$-密集となることが保証される．

頂点集合 S が上記条件を満たすとき，S は密集条件を満たすと呼ぶ．エージェント A が密集条件を満たす頂点集合 S を構成できたとき，以下の議論により準線形時間でのランデブーを達成することがすることが可能になる．

- エージェント A は $N(S)$ の頂点からいくつかを一様ランダムに選択して訪問し，エージェント B は $N(v_0^B)$ からいくつかの頂点一様ランダムに選択して訪問する．誕生日の逆理の解析と同様に，$c = |N(S) \cap N(v_0^B)|$ とすると，$N(S) \cap N(v_0^B)$ よりエージェント A，B がともに $\Theta(\sqrt{c} \log n)$ 個の頂点をランダムに選択すると，高確率である一つの頂点を両エージェントが訪問することが保証される．エージェント B は訪問先の頂点にそ v_0^B のノード ID の情報を書き残すことで，自身の位置を相手エージェントに伝えることができる．最終的には v_0^B の情報を知ったエージェント A は当該頂点に移動して，そこでランデブーが達成できる．(S, v_0^B) が $(32, \delta)$-密集であることより，エージェント A，B のランダム選択は定数確率で $N(S_i) \cap N(v_0^B)$ 内の頂点を選択するため，$c = \Theta(\delta) = \Theta(n)$ であることと併せて，全体で $\Theta(\sqrt{n} \log n)$ 回のランダムな頂点選択をすれば十分となる．1 回のランダム選択は $O(|S|)$ 時間で可能であるので，全体としてこのケースでランデブーに要するラウンド $O(|S|\sqrt{n} \log n)$ ラウンドである．

部分集合 S は，初期値 $S = \{v_0^A\}$ の状態から始めて，順

ランデブーを達成している

次頂点を追加することによって構成される．構成の途中における，$|S| = i$ であるときの S の内容を S_i で表すとする．実際の動作として，エージェント A は $N(S_i)$ への訪問と S_i への頂点の追加を繰り返し行う．つまり，S_i が所望の条件を満たしているかどうかにかかわらず $N(S_i)$ への訪問を行い，ランデブーが達成されなかったとき S_i が条件を満たしていないと判断して S_i へ頂点を追加する動作を行う．頂点追加処理の概略は以下のとおりである．

- まず，エージェント A は各頂点 $y \in N^+(v_0^A) \setminus S_i$ に対して対 (S_i, y) が $(8,\delta)$-密集かどうか判定するサブルーチンを実行する．そして，(S_i, y) が $(8,\delta)$-密集とならない任意の頂点 y をを S_i に追加する．直観的には，この動作の目的は $N(S_i)$ と共有する隣接頂点が少ない $N(y)$ を追加することで，$N(S_{i+1})$ と $\bigcup_{v \in N^+(v_0^A)} N^+(v)$ の共有頂点数を効率的に増やすことである．

以上の 2 つの動作をまとめて 1 フェーズと呼ぶことにする．S はいつか v_0^B を含むため，このアルゴリズムが停止することは明らかである．以降，t をアルゴリズムが要するフェーズ数とし，x_i をフェーズ i で S に追加される頂点とする．すなわち，$S = \{x_1, x_2, \ldots, x_t\}$ であり，$S_i = \{x_1, x_2, \ldots, x_i\}$ である．次節では，アルゴリズムの詳細について述べる．

3.2 アルゴリズムの詳細

3.2.1 エージェント B の動作

エージェント B は次の動作をランデブーが完了するまで続ける．

(1) 未訪問の $N(v_0^B)$ の頂点からランダムに頂点を選び，訪問する
(2) 訪問した頂点の白板に v_0^B の値を書き残し，v_0^B へと戻る．

3.2.2 エージェント A のフェーズ i における動作:$N(S_i)$ の訪問

エージェント A は i フェーズ目において S_{i-1} に頂点を一つ追加したあと，次の動作を $\Theta\left(\sqrt{|N^+(S_i)|}\log n\right)$ 回繰り返す．

(1) 未訪問の $N^+(S_i)$ の頂点からランダムに頂点を選び，訪問する
(2) 訪問した頂点の白板を確認する．頂点の ID が書き込まれていることを確認したとき，その頂点へと訪問し停止する．そうでない場合は v_0^A へと戻る．

上記繰り返しののち，S_{i+1} の構成のため，追加する頂点を選択する．

3.2.3 追加頂点の選択

フェーズ i において，エージェント A は現在の S_i に基づいて，v_0^A の隣接部分集合 $N(v_0^A) \setminus S_i$ の中で S_i と $(8,\delta)$-密集でない頂点 $y \in N(v_0^A) \setminus S_i$ を検出する．そのような頂点が，S_i に追加する頂点の候補となる．この頂点を検出するためのエージェント A は動作は次のようになる．まず，各頂点 $v \in V$ に対応するカウンタ $\mathtt{c} : V \to \mathbb{Z}$ を用意する．カウンタの初期値は $v \in N(v_0^A) \setminus S_i$ は 0，それ以外は -1 とする．そしてエージェントは次の動作を $640|N(S_i)|\log n/\delta$ 回繰り返す．

(1) エージェント A は $N(S_i)$ の中から一様ランダムに頂点 w を訪問する．
(2) $N^+(w) \cap \left(N(v_0^A) \setminus S_j\right)$ 内の全ての頂点のカウンタを 1 インクリメントする．
(3) v_0^A に戻る

この動作によって得られたカウンタ $\mathtt{c}$ としきい値 $40\log n$ に対して，各頂点 $y \in N(v_0^A) \setminus S_i$ に対して $\mathtt{c}[y] \le 40\log n$ となる頂点が所望の $(8,\delta)$-密集でない頂点となる．よって，エージェント A はそのような y のうち任意の 1 つを選び，それを x_{i+1} として S_i への追加を行う（$S_{i+1} = S_i \cup \{v\}$）．$(8,\delta)$-密集でない頂点が存在しないとき，そのほかの任意の頂点を一つ選び，S_i に追加する．x_{i+1} を追加する際に，エージェント A は頂点 x_{i+1} に訪問し，$N(x_{i+1})$ を記憶する．$x_{i+1} = v_0^B$ である場合にランデブーを達成するために，エージェント A は 2 ラウンドその頂点に留まる．

4. アルゴリズムの解析

このアルゴリズムが高確率で $O(\sqrt{n}\log n)$ 時間でランデブーを完了することを示す．このランデブーの高速性を示すために，i フェーズ目における S_i と δ に対して，任意の $y \in N(v_0^A) \setminus S_i$ が $(32,\delta)$-密集である場合に高確率でランデブーを完了することを示す．

補題 3. ある i フェーズにおいて，S_i と δ に対して，任意の $y \in N(v_0^A) \setminus S_i$ が $(32,\delta)$-密集であると仮定する．このとき，t_0 時刻に開始した隣接訪問アルゴリズムは十分大きな n に対して確率 $1 - 1/n^3$ 以上で時刻 $t_0 + O(\sqrt{n}\log n)$ においてランデブーを完了している．

Proof. まず，エージェント B が $64\sqrt{\delta}\log N$ 個の異なる頂点に訪問したとき，$N(S_i) \cap N(v_0^B)$ 内の異なる $\Theta(\sqrt{\delta}\log n)$ 個の頂点に訪問することを示す．各 $j \in [1, 64\sqrt{\delta}\log N]$ に対して，エージェント B が j 番目以前のの訪問の時点ですでに訪問している $N(S_i) \cap N(v_0^B)$ 内の頂点数を k_j で表す．このとき，j 番目の訪問で $N(S_i) \cap N(v_0^B)$ に訪問する確率は $\frac{|N(S_i)\cap N(v_0^B)|-k_j}{|N(v_0^B)|} \ge \frac{|N(S_i)\cap N(v_0^B)|-x_j}{\Delta}$ である．よって $64\sqrt{\delta}\log N$ 回で訪問する異なる共有頂点数の期待値は $\frac{|N(S_i)\cap N(v_0^B)|-x_j}{\Delta} \cdot 64\sqrt{\delta}\log N \ge \frac{2\delta^{1.5}\log N - 64^2\sqrt{\delta}\log N}{\Delta}$ となる．訪問する異なる頂点数を X と置くと，$\delta = \gamma n$，$\Delta \le n$ を用いると Chernoff 限界より，十分大きな n に対して $X \le (1/2)\mathbb{E}[X]$ となる確率は十分に小さく (例えば $1/n^4$ 以下に) なる．

次にエージェント A が $960\sqrt{|N(S_i)|}\log N$ 回の訪問で $\Theta(\delta/\sqrt{n})$ 個の共有頂点に訪問することを示す．エー

ジェント A が一回の訪問で共有頂点に訪問する確率は $|N(S_i, v_0^B)|/|N(S_i)| \geq \delta/(32|N(S_i)|)$ となる．よって，共有頂点に訪問する回数を確率変数 Y と置くと，$960\sqrt{|N(S_i)|}\log N$ 回繰り返したとき，X の期待値は $\mathbb{E}[X] = 30\delta \log N/\sqrt{|N(S_i)|} \geq 30\delta \log N/\sqrt{n}$ となる．$\delta = \Theta(n)$ と Chernoff 限界を用いると十分大きな n に対して $(1/2)\mathbb{E}[X]$ となる確率は十分に小さくなる，例えば $1/n^4$ 以下になる．

以上の 2 つの事象が成立すると仮定する．このとき，エージェント B が訪問した頂点にエージェント A が一度も訪問しない確率は

$$\left(1 - \frac{X}{|N(S_i) \cap N(v_0^B)|}\right)^Y \leq \exp\left(\frac{30\delta^{1.5}\log^2 N}{\Delta\sqrt{n}}\right)$$

となり，十分大きな n に対してこれは $1/n^4$ より小さくなる．$N(S_i)$ 中の任意の頂点 v が与えられたとき，エージェント A が現在地より v に訪問するためには高々 $|S_i|+1$ 回の移動で十分なため，以上より，エージェント A は時刻 $t_0 + 960|S_i|\sqrt{|N(S_i)|\log N} = t_0 + O(|S_i|\sqrt{n}\log n)$ において確率 $1 - 1/n^3$ 以上でランデブーを完了する． □

次に，S_i の構成法が少ないフェーズ数で密集条件を満たす S を構成することを示す．この証明のために，S_i の構成において，高確率で S_i と $(8,\delta)$-密集でない頂点 $y \in N(v_0^A) \setminus S_i$ を検出することが可能であることを示す．直感的には，ある頂点 $y' \in N(v_0^A) \setminus S_i$ が S_i と $(8,\delta)$-密集しているとき，$N(S_i)$ の多くの頂点は y' と隣接している．そのために，$\Theta(|S_i|\log n/\delta)$ 回 $N(S_i)$ 内の頂点に訪問したとき，y のカウンタ値が多くインクリメントされることが期待できる．一方，$y \in N(v_0^A) \setminus S_i$ が $N(S_i)$ のほとんどと隣接していないとき，y のカウンタ値はほとんどインクリメントされない．

補題 4. 確率 $1 - 1/n^4$ 以上で全ての対 (S_i, y) $(1 \leq i \leq N^+(v_0^A), y \in N(v_0^A) \setminus S_i)$ に対して，次の 2 つの命題がともに成立する：(1) (S_i, y) が $(8,\delta)$-密集であるとき，$\mathtt{C}[y] \geq 40\log n$ となる． (2) (S_i, y) が $(32,\delta)$-密集でないとき，$\mathtt{C}[y] \leq 40\log n$ となる．

Proof. 任意の $y \in N(v_0^A)$ について考える．1 回の隣接集合 $N(S_i)$ 内の頂点の訪問で，y のカウンタ値がインクリメントされる確率は $|N(S_i) \cap N(y)|/N(S_i)$ である．また，これを $640|N(S_i)|\log n/\delta$ 回繰り返したときのカウンタ値の期待値を確率変数 X で表すと，期待値は $\mathbb{E}[X] = 640|N(S_i) \cap N(y)|\log n/\delta$ となる．ここで，(S_i, y) が 8-密集であると仮定する，すなわち，$|N(S_i) \cap N(y)| \geq \delta/8$ であると仮定する．このとき，$\mathbb{E}[X] \geq 80\log n$ が成立する．Chernoff 限界より，X が $(1/2)\mathbb{E}[X]$ より小さくなる確率は $\Pr[X \leq (1/2)\mathbb{E}[X]] \leq \mathrm{e}^{-(80\log n)/8} \leq 1/n^{10}$ となる．次に，(S_i, y) が 32-密集でないと仮定する，すなわち，$|N(S_i) \cap N(y)| \leq \delta/32$ であると仮定する．この場合，$\mathbb{E}[X] \leq 20\log n$ が成立する．Chernoff 限界より，X が $2\mathbb{E}[X]$ より大きくなる確率は，$\Pr[X \geq 2\mathbb{E}[X]] \leq \mathrm{e}^{-20\log n/3} \leq 1/n^6$ となる．

以上の議論を $N(v_0^A) \setminus S_i$ の全ての頂点及び $i \in [1, n]$ に対して適用する．任意の i に対して $|N(v_0^A) \setminus S_i| \leq n$ が成立する．よって，高々 n^2 個の和集合上界を取ることで，補題の言明が $1 - 1/n^4$ 以上の確率で成立することが示される． □

上記補題の言明が高確率で正しく動作すると仮定したとき，所望の S_i が高確率で高速に構成できることを示すことができる．このとき，任意の S_i, δ に対して $(32,\delta)$-密集とならない頂点が存在しない場合，S_i は密集条件を満たしていることになり，高速なランデブーが可能になる．

補題 5. 補題 4 の命題が確率 $1 - 1/n^4$ で成立すると仮定する．このとき，少なくとも $t = 8n/(7\delta) = \Theta(1)$ について，S_t は密集条件を満たす

Proof. 補題 4 の成功性の保証をもとに，$t \geq 8n/(7\delta)$ である場合に $(32,\delta)$-密集となる対 $(S_t, y)(y \in N(v_0^A \setminus S_t))$ が存在しないことを示す．これは，$t = 8n/(7\delta)$ である場合に，$N(S_i)$ が全ての頂点 n を取り尽くすことによって示す．

任意の $i \in [1, t-1]$ に対して，x_{i+1} が追加されたとき，頂点 x_{i+1} と集合 $\{x_1, \ldots, x_i\}$ の間で，

$$\left|\bigcup_{j=1}^{i} N^+(x_j) \cap N(x_{i+1})\right| \leq \delta/8$$

が成立する．よって，新たに少なくとも $\delta - \delta/8 = (7/8)\delta$ 個の異なる頂点が $N(S_i)$ に追加される．

これが全ての $i \in [1, t-1]$ に対して適用できる．よって

$$\left|\bigcup_{i=1}^{t} N^+(x_i)\right| \geq \delta + \frac{7}{8}(t-1)\delta \geq \frac{7}{8}t\delta$$

が成立．また，自明な上界として $|\bigcup_{i=1}^{t} N(x_i)| \leq n$ が成立．これらより，

$$\frac{7}{8}t\delta \leq n$$
$$t \leq \frac{8}{7}\cdot\frac{n}{\delta}$$

が成立する．この不等式より，$|S_t| = 8n/(7\delta)$ となったとき $N(S_i) = n$ が成立し，S_i と 32-希薄となる頂点 y が存在しないことが示された． □

最後に，提案アルゴリズムが高確率で $O(\sqrt{n}\log n)$ 時間でランデブーを完了することを示す．

定理 6. 提案アルゴリズムは確率 1 でランデブーを完了し，高確率で $O(\sqrt{n}\log n)$ 時間でランデブーを完了する．

Proof. 両エージェントが確率 1 でランデブーを完了する

ことは，アルゴリズムの構造からエージェント A が最終的に $N(v_0^A)$ の全ての頂点に訪問することから明らかである．よって，証明すべきはランデブーを達成するために必要な時間が高確率で $O(\sqrt{n}\log n)$ ラウンドとなることである．補題 4 の命題は確率 $1-1/n^4$ 以上で成立する．このとき，補題 5 より $O(1)$ 個の頂点を集合 S に追加したとき，任意の頂点 $y \in N(v_0^A) \setminus S$ に対して対 (S, y) が $(32, \delta)$-密集となる．よって補題 3 より，確率 $1-1/n^3$ 以上でエージェントはランデブーを完了する．全体の実行時間は，各フェーズの実行時間が $O(t\sqrt{n}\log n)$ 時間であることから，$O(t^2\sqrt{n}\log n)$ ラウンドとなり，高確率で $t = O(1)$ であることより $O(\sqrt{n}\log n)$ ラウンドとなる． □

5. まとめと今後の課題

本稿では，最小次数が $\delta = \Omega(n)$ となるような任意のグラフ上で距離 1 上に配置された 2 エージェントが高確率で $O(\sqrt{n}\log n)$ 時間でランデブーを完了するアルゴリズムを提案し，解析を行った．今後の課題として，最小次数 $\delta = o(n)$ の場合も含めた，一般のケースにおける準線形時間ランデブーの可能性，および初期距離 2 以上の場合における可能性についての検討が挙げられる．

謝辞 本研究は，JST 戦略的国際共同研究プログラム（SICORP），ならびに JSPS 科研費 16H02878 の支援を受けている．

参考文献

[1] Abbas, S., Mosbah, M. and Zemmari, A.: A probabilistic model for distributed merging of mobile agents, 2nd international workshop on verification and evaluation of computer and communication systems (VeCOS' 08) (2008).
[2] Alpern, S.: Rendezvous search: A personal perspective, *Operations Research*, Vol. 50, No. 5, pp. 772–795 (2002).
[3] Alpern, S.: Rendezvous search on labeled networks, *Naval Research Logistics (NRL)*, Vol. 49, No. 3, pp. 256–274 (2002).
[4] Alpern, S., Fokkink, R., Gasieniec, L., Lindelauf, R. and Subrahmanian, V.: *Search theory*, Springer (2013).
[5] Alpern, S. and Gal, S.: *The theory of search games and rendezvous*, International Series in Operations Research & Management Science, Vol. 55, Springer Science & Business Media (2006).
[6] Anderson, E. J. and Weber, R.: The rendezvous problem on discrete locations, *Journal of Applied Probability*, Vol. 27, No. 4, pp. 839–851 (1990).
[7] Baba, D., Izumi, T., Ooshita, F., Kakugawa, H. and Masuzawa, T.: Linear time and space gathering of anonymous mobile agents in asynchronous trees, *Theoretical Computer Science*, Vol. 478, pp. 118–126 (2013).
[8] Bouchard, S., Dieudonné, Y., Pelc, A. and Petit, F.: On deterministic rendezvous at a node of agents with arbitrary velocities, *Information Processing Letters*, Vol. 133, pp. 39–43 (2018).
[9] Bshouty, N. H., Higham, L. and Warpechowska-Gruca, J.: Meeting times of random walks on graphs, *Information Processing Letters*, Vol. 69, No. 5, pp. 259 – 265 (1999).
[10] Collins, A., Czyzowicz, J., Gąsieniec, L., Kosowski, A. and Martin, R.: Synchronous rendezvous for location-aware agents, *International Symposium on Distributed Computing*, Springer, pp. 447–459 (2011).
[11] Czyzowicz, J., Kosowski, A. and Pelc, A.: How to meet when you forget: log-space rendezvous in arbitrary graphs, *Distributed Computing*, Vol. 25, No. 2, pp. 165–178 (2012).
[12] Czyzowicz, J., Kosowski, A. and Pelc, A.: Time versus space trade-offs for rendezvous in trees, *Distributed Computing*, Vol. 27, No. 2, pp. 95–109 (2014).
[13] Czyzowicz, J., Pelc, A. and Labourel, A.: How to meet asynchronously (almost) everywhere, *ACM Transactions on Algorithms (TALG)*, Vol. 8, No. 4, p. 37 (2012).
[14] Dani, V., Hayes, T. P., Moore, C. and Russell, A.: Codes, lower bounds, and phase transitions in the symmetric rendezvous problem, *Random Structures & Algorithms*, Vol. 49, No. 4, pp. 742–765 (2016).
[15] Das, S., Dereniowski, D., Kosowski, A. and Uznański, P.: Rendezvous of distance-aware mobile agents in unknown graphs, *International Colloquium on Structural Information and Communication Complexity*, Springer, pp. 295–310 (2014).
[16] De Marco, G., Gargano, L., Kranakis, E., Krizanc, D., Pelc, A. and Vaccaro, U.: Asynchronous deterministic rendezvous in graphs, *Theoretical Computer Science*, Vol. 355, No. 3, pp. 315–326 (2006).
[17] De Marco, G., Gargano, L., Kranakis, E., Krizanc, D., Pelc, A. and Vaccaro, U.: Asynchronous deterministic rendezvous in graphs, *Theoretical Computer Science*, Vol. 355, No. 3, pp. 315–326 (2006).
[18] Dereniowski, D. and Pelc, A.: Drawing maps with advice, *Journal of Parallel and Distributed Computing*, Vol. 72, No. 2, pp. 132–143 (2012).
[19] Flocchini, P., Kranakis, E., Krizanc, D., Santoro, N. and Sawchuk, C.: Multiple mobile agent rendezvous in a ring, *Latin American Symposium on Theoretical Informatics*, Springer, pp. 599–608 (2004).
[20] Fraigniaud, P. and Pelc, A.: Deterministic rendezvous in trees with little memory, *International Symposium on Distributed Computing*, Springer, pp. 242–256 (2008).
[21] Kranakis, E., Krizanc, D. and Rajsbaum, S.: Mobile agent rendezvous: A survey, *International Colloquium on Structural Information and Communication Complexity*, Springer, pp. 1–9 (2006).
[22] Kranakis, E., Santoro, N., Sawchuk, C. and Krizanc, D.: Mobile agent rendezvous in a ring, *Distributed Computing Systems, 2003. Proceedings. 23rd International Conference on*, IEEE, pp. 592–599 (2003).
[23] Miller, A. and Pelc, A.: Fast rendezvous with advice, *Theoretical Computer Science*, Vol. 608, pp. 190–198 (2015).
[24] Miller, A. and Pelc, A.: Tradeoffs between cost and information for rendezvous and treasure hunt, *Journal of Parallel and Distributed Computing*, Vol. 83, pp. 159–167 (2015).
[25] Miller, A. and Pelc, A.: Time versus cost tradeoffs for deterministic rendezvous in networks, *Distributed Computing*, Vol. 29, No. 1, pp. 51–64 (2016).
[26] Pelc, A.: Deterministic rendezvous in networks: A comprehensive survey, *Networks*, Vol. 59, No. 3, pp. 331–347 (2012).

ⓒ 2018 Information Processing Society of Japan

[27] Tetali, P. and Winkler, P.: On a Random Walk Problem Arising in Self-stabilizing Token Management, *Proceedings of the Tenth Annual ACM Symposium on Principles of Distributed Computing*, PODC '91, New York, NY, USA, ACM, pp. 273–280 (1991).
[28] Weber, R.: Optimal symmetric rendezvous search on three locations, *Mathematics of Operations Research*, Vol. 37, No. 1, pp. 111–122 (2012).
[29] Yu, X. and Yung, M.: Agent rendezvous: A dynamic symmetry-breaking problem, *International Colloquium on Automata, Languages, and Programming*, Springer, pp. 610–621 (1996).

ⓒ 2018 Information Processing Society of Japan