

# モバイルロボット群による重み付きグラフ上でのペアワイズ 合流問題

池田 亮介<sup>1,a)</sup> 山内 由紀子<sup>2,b)</sup>

**概要:** 本研究では、頂点重みつきグラフにおけるモバイルロボット群のペアワイズ合流問題を提案する。頂点の重みは、各頂点に同時に存在することのできるロボットの最大の数を表す。2台のロボットがある時点で同じ頂点に同時に存在するとき、その2台のロボットは合流したという。ペアワイズ合流問題とは、各ロボットが自身以外のロボットすべてと合流する問題である。本研究では、2分木において頂点に少数のロボットを集合させることを繰り返してペアワイズ合流問題を解く手法を示す。

**キーワード:** アルゴリズム論, 分散アルゴリズム, モバイルロボット

## Pairwise meeting problem on weighted graphs with a group of mobile robots

**Abstract:** We propose the **pairwise meeting problem** for a group of mobile robots in a vertex-weighted graph. The vertex weight represents the maximum number of robots that can simultaneously reside at the vertex. Two robots are said to **meet** if they are simultaneously present at the same vertex at a given time. The pairwise meeting problem requires that each robot meets all the other robots. In this study, we present a method for solving pairwise meeting problems for a binary tree by repeatedly gathering a small number of robots at a vertex.

**Keywords:** Algorithm theory, distributed algorithms, mobile robots

### 1. はじめに

近年、自律的に移動するモバイルロボットが工学、理論計算機科学、さらに運送や環境観測などでも注目を集めている。例えば、ルンバなどの掃除ロボットや倉庫で荷物を搬送するロボットなどが挙げられる。モバイルロボット群の分散協調理論に関する研究も多数行われており、モバイルロボットの機能と問題の可解性の関係が議論されている。これらの研究では特に、記憶領域や計算能力が乏しく、区別が不可能(匿名)であり、互いに通信を行うことが不可能であるという低機能なロボットを想定することが多い。

既存研究の多くでは、モバイルロボット群が移動する空間は、連続空間や離散空間としてモデル化とされてきた。本研究では、離散空間、つまりグラフ上を移動するモバイ

ルロボットを想定する。ロボットは、観測 (*Look*)、計算 (*Compute*)、移動 (*Move*) の3つのフェーズで構成されるサイクルを繰り返し実行する。観測フェーズでは、ロボットはグラフの形状や各ロボットの位置を観測結果として取得する。計算フェーズでは、観測結果をもとにして共通のアルゴリズムを用いて移動先を決定する。ロボットが無記憶とは、ロボットはメモリを持っておらず、観測結果のみに基づいて移動先を決定することをいう。移動フェーズでは、計算フェーズで決定した移動先に実際に移動する。この観測、計算、移動フェーズの長さが決まっており、すべてのロボットが観測、計算、移動フェーズを同時に始めるモデルを完全同期モデルという。一方、観測、計算、移動フェーズの長さが有限であるが決まっておらず、すべてのロボットが観測、計算、移動フェーズを任意のタイミングで始めるモデルを非同期モデルという。

グラフ上を動くモバイルロボットに関しては様々な分散協調タスクが考えられてきた。探索問題とは、全ての頂点

<sup>1</sup> 九州大学工学部電気情報工学科

<sup>2</sup> 九州大学大学院システム情報科学研究院

<sup>a)</sup> ikeda.ryosuke.219@s.kyushu-u.ac.jp

<sup>b)</sup> yamauchi@inf.kyushu-u.ac.jp

をいずれかのロボットが訪問する問題である。Flocchiniらはリング状のグラフで、非同期な無記憶なロボット群による探索問題の解法を提案した [1]。この論文では、グラフの頂点数  $n$  とロボットの総数  $k$  が互いに素であり、かつ  $k \geq 17$  の場合、探索問題が解けることが示されている。**集合問題**とは、全てのロボットが1つの頂点に集まる問題である。集合問題はロボットの配置やグラフの構造から、集合する頂点を決定し、その頂点に全てのロボットが集合することで解くことができる。グラフの中心がただ1つに定まる場合などは、ロボットが容易に集合する頂点に合意し、その頂点に移動して集合することができる。リングやグリッドといった対称性が高いグラフにおいてもロボットの初期配置、つまりどの頂点にロボットがいるかによって、集合問題を解くことができる場合がある。Klasingらは、非同期な無記憶なロボット群による初期配置が対称な場合の集合問題の解法を提案した [2]。ロボットの初期配置の対称性は初期配置に対する自己同型写像としてとらえることができる。代表的なものが、リング状のグラフの初期配置に対する回転対称性(回転中心に対する対称性)や鏡映対称性(軸に対する対称性)である。この論文では、ロボットの総数  $k$  が  $k > 18$  かつ、初期配置の対称軸が1つしかない場合について集合問題が解けることが示されている。 $k$  が奇数なので、初期配置が対称でも集合する頂点を対称軸上に決定することができる。D'Angeloらはグリッドやツリーで、非同期、無記憶で識別不能なロボット群による集合問題の解法を提案した [3]。この論文ではグリッドについて、縦と横の長さがともに奇数の場合と片方が偶数の場合について、グリッドの縦と横を通る中心線を利用して集合する頂点を決定している。ツリーについては、ツリーの中心と呼ばれる頂点を集合する頂点に決定している。このように、集合問題はロボットの配置やグラフの構造から、集合する頂点を決定し、その頂点に全てのロボットが集合する。

しかし、現実のモバイルロボットでは、物理的制限があるため、同じ場所に複数のモバイルロボットは存在できない可能性がある。もし、集合しようとした頂点に全てのロボットが存在することができないなら、集合問題を解くことができない。そこで、頂点に存在するロボット数に制限がある場合の集合問題に代わる問題を考える必要がある。

本研究では頂点重みつきグラフにおける**ペアワイズ合流問題**を提案する。頂点の重みは、各頂点に同時に存在することのできるロボットの最大の数を表す。2台のロボットがある時点で同じ頂点に同時に存在するとき、その2台のロボットは**合流した**という。**ペアワイズ合流問題**とは、各ロボットを他のロボットすべてと合流させる問題である。重みがロボットの台数以上の頂点が存在する場合は、その頂点に全てのロボットを集合させることでペアワイズ合流問題を解くことができるが、すべての頂点の重みがロボッ

トの台数未満の場合は集合によりペアワイズ合流問題を解くことはできない。本研究では、頂点に少数のロボットを集合させることを繰り返してペアワイズ合流問題を解く手法を与える。

本稿では、まずモバイルロボットモデルとペアワイズ合流問題を定義し、無記憶なロボットによるペアワイズ合流問題の不可能性に関する定理をいくつか示す。次に、2分木において有記憶なロボットがペアワイズ合流問題を解くためのアルゴリズムを与える。

## 2. 準備

### 2.1 基本的な用語と定義

頂点重みつき無向グラフ  $G = (V, E)$  上を自律的に移動する**識別不能な  $k$  台のモバイルロボット**の集合  $R = \{r_1, r_2, \dots, r_k\}$  を想定する。 $r_i$  という表記は説明のため使用する。 $|V| = n$  とし、 $k < n$  とする。時刻  $t$  に  $r_i$  が存在する頂点を  $p_i(t)$  と表す。時刻  $t$  に  $r_i$  と同じ頂点  $p_i(t)$  に存在するロボット群を  $R'_i(t) \subseteq R$  と表す。多重集合  $P(t) = \{p_1(t), p_2(t), \dots, p_k(t)\}$  を  $R$  の時刻  $t$  での**配置**という。

各ロボット  $r_i$  は**観測**、**計算**、**移動**という3つのフェーズから成る**サイクル**を繰り返し実行する。観測フェーズでは、 $r_i$  は観測フェーズ中のある時点でグラフのどの頂点に何台のロボットが存在するかを得る。ただし、各ロボットはグラフの頂点を識別できないので、時刻  $t$  における観測結果は  $G$  の何らかの自己同型写像を  $f$  とすると、配置  $\{f(p_1(t)), f(p_2(t)), \dots, f(p_k(t))\}$  となる。よって、観測結果はロボット  $r_i$  と  $r_j$  ( $r_i \neq r_j$ ) で一致するとは限らない。計算フェーズでは、 $r_i$  は直前の観測結果をもとに、現在いる頂点ににとどまるか、とどまらないなら現在いる頂点のどの隣接頂点に移動するかを共通の決定性アルゴリズム  $A$  を用いて計算する。移動フェーズでは、 $r_i$  は直前の計算で求めた頂点に移動する。この移動にかかる時間は一瞬であり、1回の移動で0または1つの辺を移動する。

**完全同期モデル**では、観測、計算、移動フェーズの長さが決まっており、全てのロボットが同じタイミングで、観測、計算、移動フェーズを行う。**非同期モデル**では、観測、計算、移動フェーズの長さは有限であるが決まっておらず、各ロボットが観測、計算、移動フェーズを始めるタイミングは任意であるが各ステップ、つまりサイクルの長さは有限である。各ロボットは無限回サイクルを実行するとする。本稿では、非同期モデルを想定する。

$v$  の重み  $w(v)$  は共通で  $c$  であり、1つの頂点には同時に  $c$  台までしかロボットが存在できないとする。さらに、ある頂点に同時に複数の接続辺から多数のロボットが移動してきて、その頂点に存在するロボットの台数が  $c$  を超える場合、移動を行おうとしたロボットはすべて移動できず、現在いる頂点にとどまるとする。移動できなかったロ

ポットについては、移動できなかったことを認識できないとする。

本稿では、 $n$  個の頂点からなる 2 分木を想定する。2 分木とは、各頂点の次数が、高々 3 であるような連結でサイクルがないグラフである。

頂点  $v_i \in V$  の次数を  $\delta_i$  と表す。  $v_i$  に接続する各辺について、**ポート番号**として、 $\{0, 1, \dots, \delta_i - 1\}$  に含まれる数が重複することなく割り当てられるとする。ポート番号の割り当て方は、各頂点で任意となっている。ロボットは移動を行うと、移動した頂点にどのポートから入ってきたかを認識できるものとする。

時刻  $t$  に頂点  $v_i$  に存在するロボットの台数を  $d_i(t)$  と表す。  $d_i(t) > 1$  のとき、  $v_i$  を**多重点**と呼ぶ。ロボットの配置の対称性を考える。グラフ  $G$  上のロボットの配置  $P$  に、  $\forall v \in V$  に対して、頂点  $v$  と  $f(v)$  にいるロボット数が一致するような  $G$  の自己同型写像  $f: V \rightarrow V$  のみが存在するとき、  $P$  は**対称**であるという。そのような  $G$  の自己同型写像が存在しないとき、  $P$  は**非対称**であるという。

パスグラフにおけるロボットの初期配置が対称とは、  $n$  が偶数のとき、  $1 \leq i \leq \frac{n}{2}$  を満たす任意の  $i$  に対して  $d_i(0) = d_{n-i}(0)$  が、  $n$  が奇数のとき、  $1 \leq i \leq \frac{n-1}{2}$  を満たす任意の  $i$  に対して、  $d_i(0) = d_{n-i+1}(0)$  が成立することである。

初期配置  $P(0)$  から非同期モデルのロボット群がアルゴリズム  $A$  を実行する時、いずれかのロボットが観測を行った時刻を  $t_1, t_2, \dots$  とする。  $t_i$  を時刻  $i$  とみなした時、  $P(0), P(1), P(2), \dots$  を  $A$  の**実行**と呼ぶ。非同期モデルでは、ロボットがいずれの時刻にサイクルや個々のフェーズを実行したかにより、ひとつの初期配置からでも多数の実行が存在する。

**合流問題** とは各ロボットを他の全てのロボットと合流させる問題である。ここで、ある実行  $P(0), P(1), P(2), \dots$  において、ロボット  $r_i$  と  $r_j$  が合流するとは、  $p_i(t) = p_j(t)$  となる  $t$  が存在することである。ロボットの部分集合  $R' \subseteq R$  が合流するとは、  $R'$  に含まれる任意の  $r_i, r_j$  が合流することである。同じ頂点に複数のロボットが存在しないような初期配置  $P(0)$  から始まるアルゴリズム  $A$  の任意の実行  $P(0), P(1), P(2), \dots$  について、有限の  $T \geq 1$  が存在し、各ペア  $r_i, r_j \in R (i \neq j)$  について、  $p_i(t) = p_j(t)$  を満たす  $1 \leq t \leq T$  が存在するとき、  $A$  は初期配置  $P(0)$  から**合流問題を解く** という。

与えられた 2 分木  $G = (V, E)$  から根を一意に決定するために必要な**中心**を定義する。まず、頂点  $v$  における**離心率**  $\epsilon(v)$  とは、  $v$  から最も離れた頂点  $w$  までの距離である。すなわち、  $\epsilon(v) = \max d(v, w) (w \in V, w \neq v)$  と表される。グラフの中心とは、離心率が最小になる頂点である。中心は、グラフの構造から決定することができ、木グラフに必ず 1 つもしくは 2 つ存在する。

## 2.2 無記憶ロボットのペアワイズ合流問題

まず、ペアワイズ合流問題はロボットが無記憶で、  $k > c$  の場合、決定性アルゴリズムでは解けないことを示す。

**補題 2.1** 頂点重みつきグラフ上を移動する非同期な無記憶なロボット群を想定する。任意のアルゴリズムについて、時刻  $t$  に、ある頂点に  $R' \subseteq R$  が存在するとき、任意の  $t' > t$  で  $R'$  に属するロボットが全て同じ頂点に存在するような非同期実行が存在する。

**証明.** 全てのロボットが完全に同期してアルゴリズムを実行する場合を考える。任意の  $t' > t$  について、  $R'$  に属するロボットは同時に観測を開始し、同じ観測結果を得るので、共通のアルゴリズムを用いて、移動先を決定する。ロボットは無記憶であることから、移動先は観測結果のみに基づき決定されるので、同じ観測結果を得たロボットの移動先は同じになる。ここで、  $R'$  に属するロボットが同時に頂点  $u_i$  に移動先を決定したとき、  $u_i$  にいるロボットと  $R'$  に属するロボットの総数が  $c$  台を超えない場合と超える場合について考える。まず、  $u_i$  にいるロボットと  $R'$  に属するロボットの総数が  $c$  台を超えないとき、  $R'$  に属する全てのロボットが重み  $c$  に関係なく  $u_i$  に移動することができる。よって、移動後も  $R'$  に属するロボットは同じ頂点に存在する。次に、  $u_i$  にいるロボットと  $R'$  に属するロボットの総数が  $c$  台を超えると、重み  $c$  の定義より、  $u_i$  へ移動しようとした  $R'$  に属するロボットは、全て移動できず現在いる頂点にとどまる。よって、  $R'$  に属するロボットは全て移動前の頂点に存在しているので、同じ頂点に存在するといえる。

以上より、すべてのロボットが同時に観測、計算、移動をするような同期実行では、任意の  $t' > t$  で  $R'$  に属するロボットは全て同じ頂点に存在することを満たす。 □

**補題 2.2** 頂点重みつきグラフ上を移動する非同期な無記憶なロボット群を想定する。このとき、任意のアルゴリズムについて、任意の  $t' > t$  で、任意のロボット  $r_i$  について  $R'_i(t') \supseteq R'_i(t)$  を満たすような非同期実行が存在する。

**証明.** 時刻  $t$  で、ロボットの集合  $R'_i(t)$  がロボットの集合  $R'' (R'' \cap R'_i(t) = \emptyset)$  と合流する場合を考える。  $|R'_i(t)| + |R''| \leq c$  のとき、  $R'_i(t)$  に属するロボットと  $R''$  に属するロボットは同じ頂点に存在する。よって、補題 2.1 より  $R'(t') = R'_i(t) \cup R'', (t' > t)$  となる。  $|R'_i(t)| + |R''| > c$  のとき、  $R'_i(t)$  に属するロボットと  $R''$  に属するロボットは合流できない。よって、補題 2.1 より、  $R'(t') = R'_i(t), (t' > t)$  となる。ゆえに、  $t' > t$  で  $R'_i(t') \supseteq R'_i(t)$  が成立する。 □

**定理 2.1** 非同期な無記憶なロボット群は、  $k > c$  の場合、ペアワイズ合流問題を解くことができない。

**証明.** ペアワイズ合流問題を解く決定性アルゴリズムを  $A$  とする。  $A$  はロボット群が同時に観測、計算、移動をする同期実行を行う場合に解けなくなることを示す。ロボット群が無記憶かつ同期実行の場合は、  $A$  の実行は、ロボット

の初期配置によってのみ決まる。ここで、任意の初期配置  $P(0)$  からの  $A$  の実行を考える。このとき、補題 2.2 より、あるロボット  $r_i$  の  $R'_i(t)$  は  $R'_i(0) \subseteq R'_i(1) \subseteq \dots \subseteq R'_i(T)$  となる。  $R'_i(t) \neq R \setminus \{r_i\}$  のとき、  $r_i$  はまだ合流していないロボットが存在する。よって、  $R'_i(t)$  は  $A$  がペアワイズ合流問題を解く時刻  $T$  において、  $R'_i(t) = \{r_1, r_2, \dots, r_k\}$  となり、全てのロボットが  $p_i(t)$  に存在するはずである。しかし、  $k > c$  の場合、1つの頂点に  $c$  台までしかロボットがいることができないという条件を満たさなくなってしまう。ゆえに  $k > c$  の場合、背理法より、ペアワイズ合流問題を解く決定性アルゴリズムは存在しない。

□

次に、ペアワイズ合流問題はロボットが無記憶でも、特定のグラフで、  $k \leq c$  の場合に解くことができることを示す。

**定理 2.2**  $k \leq c$  ならば、中心が1つの任意のグラフ  $G$  について、非同期的無記憶ロボット群で、ペアワイズ合流問題を解くことができる。

**証明.** 中心が1つ存在する任意のグラフにおいて、ペアワイズ合流問題を解く決定性アルゴリズムを示す。このアルゴリズムでは、ロボット  $r_i$  は観測結果から中心と現在いる頂点の最短経路を計算し、最短経路上に存在する隣接頂点に移動するという動作を中心に到達するまで続ける。このアルゴリズムを全てのロボットで行うと、中心に全てのロボットが集合するので、ペアワイズ合流問題を解くことができる。

□

### 3. 2分木におけるペアワイズ合流問題

本章では、2分木について、ペアワイズ合流問題を解く分散アルゴリズムを提案する。提案アルゴリズムでは想定している状況は  $c \geq 3$  で、以下の2つのグラフの形である。(i) 2分木の中心が1頂点で、その1頂点に対して2分木が非対称な場合。(ii) 2分木の中心が2頂点で、その2頂点を結ぶ辺に対して2分木が非対称な場合。初期配置は、多重点が存在しないと仮定する。

提案アルゴリズムでは、まずグラフの中心に3台のロボットによる多重点を作成する。次に、中心に近いロボットから順に中心に向かって移動することで、全てのロボットが存在する頂点が2分木の中心を含む連結部分グラフになるようにする。最後に、中心にいる3台のロボットのうち2台が、作成した連結グラフ内を深さ優先探索を行うことで他のロボットと合流する。探索を行う2台のロボットは、連結グラフ内の全ての頂点を探索すると、2台のうちの1台は、探索をやめる、もう1台は探索を終了した点にいるロボットと再び深さ優先探索を行う。

#### 3.1 アルゴリズム中で用いる用語の定義

与えられた2分木  $G = (V, E)$  から、根  $v_{root}$  を一意に決定する方法を考える。まず、中心が1つなら根  $v_{root}$  とし

て中心が選ばれる。次に、中心が2つある場合を考える。2つの中心を  $u, v$  とすると、  $u, v$  は隣接しているので、辺  $\{u, v\} \in E$  を取り除いて得られる2つの木のうち、  $u$  を根とした根付き木を  $T_u$ 、  $v$  を根とした根付き木を  $T_v$  とし、  $T_u$  と  $T_v$  の構造の違いを利用することで根  $r$  を1つに決定する。

根付き木の構造を文字列にする  $AHU$  アルゴリズム [4] を紹介する。このアルゴリズムでは、まず、葉に文字列 10 を割り当てる。深さ  $k$  の頂点  $u_k$  について、  $u_k$  の子供に割り当てられている文字列を辞書式順序に昇順で並び替え、その順に文字列を結合する。そして、結合した文字列の前に 1 を、後ろに 0 を結合した文字列を  $u_k$  に割り当てる。最後に、根に割り当てられた文字列を出力し、アルゴリズムが終了する。Algorithm 1 に  $AHU$  アルゴリズムを示す。はわち

---

#### Algorithm 1 $AHU(u_i)$ [4]

---

```
1: if  $u_i$  が葉ノード then
2:   return 10
3: else
4:    $temp = (w \in child(u_i)$  の  $AHU(w)$  を辞書式順の昇順に並び、連結する)
5: end if
6: return  $1temp0$ 
```

---

$T_u, T_v$  に対して、  $u$  または  $v$  を入力として  $AHU$  アルゴリズムを行った結果を  $AHU(u)$  と  $AHU(v)$  とする。提案アルゴリズムの仮定より、与えられた2分木は中心が2つある場合、2つの中心を結ぶ辺に対して非対称なので、  $T_u$  と  $T_v$  が同型ではない。提案アルゴリズムの仮定より、  $AHU(u) \neq AHU(v)$  となるので、  $AHU(u)$  と  $AHU(v)$  を辞書式順序で比較して、大きい方を根  $v_{root}$  とする。ここで、頂点  $u, v$  について、  $AHU(u)$  が  $AHU(v)$  より辞書的に大きいことを  $AHU(u) > AHU(v)$  と表し、  $AHU(u)$  が  $AHU(v)$  より辞書的に小さいことを  $AHU(u) < AHU(v)$  と表す。

図 1 に、中心が2個ある木グラフで、  $T_u$  と  $T_v$  のそれぞれに  $AHU$  アルゴリズムを行った例を示す。各頂点に書かれている文字列は、  $AHU$  アルゴリズムの実行中に割り当てられた文字列を表す。  $AHU(u)$  の方が大きいので  $v_{root}$  として  $u$  が選ばれる。与えられた2分木  $T = (V, E)$  を前述の方法で選んだ頂点  $v_{root}$  を根とする根付き木と見なす。

根  $v_{root}$  には隣接頂点が2つまたは、3つ存在する。よって、  $v_{root}$  の隣接頂点を、隣接頂点が3つのときは  $v_{c_1}, v_{c_2}, v_{c_3}$  と呼び、2つのときは  $v_{c_1}, v_{c_2}$  と呼ぶ。そして、  $v_{c_1}, v_{c_2}, v_{c_3}$  を根とする  $T$  の部分木のことをそれぞれ  $T_1, T_2, T_3$  と呼ぶ。ロボットは  $T_1, T_2, T_3$  に  $AHU$  アルゴリズムを適用し、  $AHU(v_{c_1}) \geq AHU(v_{c_2}) \geq AHU(v_{c_3})$  となるように、  $T_1, T_2, T_3$  を判別する。  $T_1, T_2, T_3$  のうち、ロボットが存在

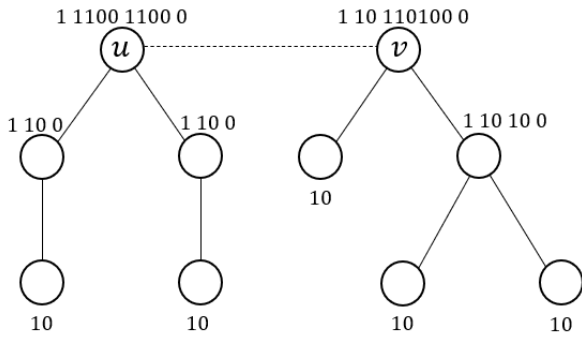


図 1 AHU アルゴリズムの実行例. 頂点に並記した 01 の文字列が AHU アルゴリズムがその頂点に与えた文字列.

する部分木の数を空ではない部分木の数と呼ぶ.

この探索をセンターブロックに含まれる頂点が根  $v_{root}$  だけになるまで行う.

#### 4. 提案アルゴリズム

提案アルゴリズムは、タワー生成フェーズ、合流フェーズの順で行う. タワー生成フェーズでは、根  $v_{root}$  にサイズ 3 のタワーを作成する. 合流フェーズでは、まず根  $v_{root}$  に存在するサイズ 3 のタワーを含むセンターブロックを作成する. 次に、サイズ 3 のタワーに存在する 3 台のロボットのうち、2 台がセンターブロック内の全ての頂点を深さ優先探索する. 2 台がセンターブロック内の全ての頂点を探索すると、2 台のうち 1 台はセンターブロックから離れ、アルゴリズムを終了するが、もう 1 台は探索を終了した点にいるロボットと再びセンターブロックの全ての頂点を探索する.

提案アルゴリズムでは、タワー生成フェーズについて、ロボットの配置に加えて空ではない部分木がいくつあるかに応じて、移動するロボットと移動先が変化する. これは、タワー生成フェーズで  $v_{root}$  にサイズ 3 のタワーを作成する時、 $r$  の隣接頂点のうち、いくつの頂点からロボットが入ってくる可能性があるかによって、サイズ 3 のタワーの作り方が変化するからである. 例えば、 $v_{root}$  にサイズ 3 のタワーを作成するとき、空ではない部分木が 1 つであれば、1 つの木から 1 台ずつロボットを根  $v_{root}$  に移動させることができ、空ではない部分木が 3 つであれば、 $v_{root}$  の 3 つの隣接頂点からそれぞれ 1 台ずつロボットを  $v_{root}$  に移動させることができる.

ロボット  $r_i$  の局所変数は以下の通りである.

- $pn_i$ . 現在いる頂点にどのポート番号の辺から入ったかが移動を終えるたびに自動的に記録される. 初期値は、 $-1$ .
- $mode_i$ . センターブロックを作成した後、どのロボットがセンターブロック内を探索するかを示す変数.  $mode_i = 0$  のロボットは、根  $r$  にタワーを作成する動作やセンターブロックを作成する動作は行うが、探索は

行わない.  $mode_i = 1$  のロボットは、センターブロックを探索する.  $mode_i = 2$  のロボットは、 $mode_i = 1$  のロボットについていくことで探索を行う.  $mode_i = -1$  のロボットは、アルゴリズムを終了し停止している.  $mode_i$  の初期値は 0.

- $vcount_i$ . センターブロックの探索中に通過した頂点数を記録する変数. 初期値は 0 とし、 $mode_i = 1$  のロボットがセンターブロック内の全ての頂点を訪問したかどうかを判断するために用いる.  $vcount_i$  とセンターブロックサイズが一致した時、 $mode_i = 1$  のロボットは自身以外の全てのロボットとの合流しているので、探索を終了する.
- $exflag_i$ . 初期値は、 $false$  とする.  $mode_i = 1$  のロボットは  $exflag_i = true$  なら探索を始めることができる. この  $exflag_i$  は、タワー生成フェーズから合流フェーズに移行した時に  $mode_i = 1$  のロボットがセンターブロックサイズが  $(k-2)$  になるまで、探索を始めないために用いられる. よって、 $mode_i = 1$  のロボットが交代するタイミングでは、 $exflag_i$  は無条件に  $true$  に変更される.
- $cblocksize_i$ . センターブロックサイズを記録するために用いる変数. 初期値は、 $(k-2)$ . 合流フェーズで最初に探索を始める  $mode_i = 1$  のロボットがセンターブロックサイズが  $k-2$  に一致するかどうかを確認することに用いられる.
- $afterflag_i$ . ロボットが移動後に、 $pn_i$  を用いて、 $mode_i$  を変更する場合に用いる  $bool$  型の変数. 初期値は、 $false$ .
- $route_i$ . スタック型の変数で、要素は、 $0, 1, 2$  からなる集合である.  $mode_i = 1$  のロボットがセンターブロックを探索するとき、通過した辺のポート番号を記録しながら深さ優先探索をするために用いる. 最初、 $route_i$  には空とする.  
 $route_i$  を操作するための関数を定義する.
  - $route_i.top()$ .  $route_i.top()$  は、 $route_i$  の一番上にある要素を返す.
  - $route_i.pop()$ .  $route_i$  の一番上にある要素を削除し、その要素を返す.
  - $route_i.push(x)$ .  $route_i$  の一番上に新たに  $x$  を追加する.

#### 4.1 タワー生成フェーズ

タワー生成フェーズでは、根  $v_{root}$  にサイズ 3 のタワーを作成する. 次に、タワーを構成する 3 台のロボットの  $mode$  をそれぞれ  $0, 1, 2$  に設定する. タワー生成は、空ではない部分木の個数によって以下の 3 つの場合に分けられる. 本稿では、アルゴリズムの概要のみを述べる.

根  $v_{root}$  にロボットが移動する過程で、同じ親をもつロ

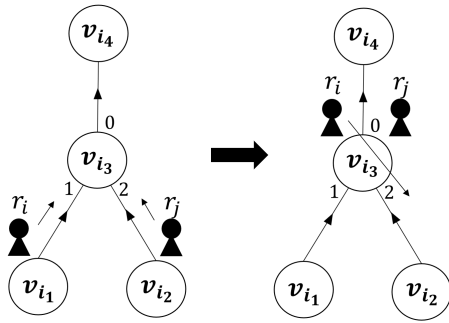


図 2 サイズ 2 のタワーの解消例

ロボットが同時に根  $v_{root}$  方向に移動したときにサイズ 2 のタワーができる場合がある。この場合、移動してきたロボット  $r_i$ ,  $r_j$  はそれぞれ,  $(pn_i + 1) \bmod 3$  と  $(pn_j + 1) \bmod 3$  のポート番号の先にある頂点を確認する。その頂点が親頂点なら現在いる頂点にとどまり、子頂点なら実際にその頂点に移動する。図 2 に、同じ親をもつロボットが同時に根  $v_{root}$  方向に移動したときにできたサイズ 2 のタワーを解消する例を示す。ロボット  $r_i, r_j$  が同時に頂点  $v_{i_3}$  移動したとき,  $pn_i = 0, pn_j = 2$  である。よって,  $(pn_i + 1) \bmod 3 = 1$ ,  $(pn_j + 1) \bmod 3 = 0$  となるため,  $r_j$  が現在いる頂点にとどまり,  $r_i$  が頂点  $v_{i_2}$  へ移動する。

まず、空ではない部分木の数が 1 個のとき、その部分木を  $T_1$  とすると、 $v_{root}$  には、 $v_{c_1}$  からしかロボットが入ってくるができない。よって、ロボットを根  $v_{root}$  に近い順に  $v_{root}$  にタワーを作成しないように移動させると、同時に  $v_{root}$  に入ってくるロボットは最大で 1 台である。よって、根  $v_{root}$  に到着した順に  $mode_i$  を割り当てることで、 $mode_i$  の値がそれぞれ 0, 1, 2 の 3 台のロボットによるタワーを作成することが可能である。

次に、空ではない部分木の数が 3 つの場合、 $T_1, T_2, T_3$  のうち同型なものが存在しないなら、一般性を失うことなく、 $AHU(v_{c_1}) > AHU(v_{c_2}) > AHU(v_{c_3})$  を満たすものとする。 $T_1, T_2, T_3$  のうち同型なものが存在するなら、その 2 つの部分木を  $T_1, T_2$  とし、残りを  $T_3$  とする。概要としては、まず根  $v_{root}$  にロボットがない場合は  $T_3$  にいるロボットを 1 台  $v_{root}$  に移動させる。次に、 $T_1$  と  $T_2$  にいるロボットをそれぞれ  $v_{c_1}$  と  $v_{c_2}$  に移動させる。最後に、 $v_{c_1}$  と  $v_{c_2}$  にいるロボットを根  $v_{root}$  に移動させる。 $mode_i$  は、 $v_{c_3}$  から根  $r_{root}$  に移動したロボットは 0 となり、 $v_{c_1}$  と  $v_{c_2}$  から根  $r_{root}$  に移動したロボットは  $(pn_i + 1) \bmod 3$  のポート番号の先の頂点が  $v_{(c_3)}$  なら 1 そうでないなら 2 となる。

最後に、空ではない部分木の数が 2 つの場合、それらを  $T_1, T_2$  とする。 $T_1$  と  $T_2$  が同型ではない場合は、 $AHU(v_{c_1}) \neq AHU(v_{c_2})$  を満たす。一般性を失うことなく、 $T_1, T_2$  は  $AHU(v_{c_1}) > AHU(v_{c_2})$  を満たすものとする。このとき、 $T_1$  にいるロボットが  $v_{root}$  に近い順に根  $v_{root}$  に移動し、 $v_{root}$  に到達した順に  $mode_i$  を割り当てる。 $T_1$  に

いるロボットが 3 台未満で  $v_{root}$  にサイズ 3 のタワーができる前に  $T_1$  にいるロボットがいなくなると空ではない部分木数が 1 つの場合となり、 $T_2$  にいて根  $v_{root}$  に近い順にロボットが根  $v_{root}$  に移動し、 $v_{root}$  に到達した順に  $mode_i$  を割り当てる。

空ではない部分木の数が 2 個で、 $T_1$  と  $T_2$  が同型な場合は、初期配置で  $v_{root}$  にロボットがいる場合といない場合にそれぞれ別の手順で、サイズ 3 のタワーを作成する。

初期配置で根  $v_{root}$  にロボットがいるとき、 $T_1$  と  $T_2$  で最も  $v_{root}$  に近いロボットがそれぞれ  $v_{c_1}$  と  $v_{c_2}$  に移動し、 $v_{c_1}$  と  $v_{c_2}$  にロボットが揃うとそれらのロボットは、 $v_{root}$  に移動する。 $mode_i$  は、元々  $r_{root}$  にいたロボットは 0 となり、 $v_{c_1}$  と  $v_{c_2}$  から根  $r_{root}$  に移動したロボットは  $(pn_i + 1) \bmod 3$  のポート番号の先の頂点が  $v_{(c_3)}$  なら 1 そうでないなら 2 をとる。

初期配置で根  $v_{root}$  にロボットがないとき、 $v_{root}$  に最も近いロボットが  $v_{root}$  に移動する。 $v_{root}$  にロボットが配置されると、全てのロボットは初期配置で  $v_{root}$  にロボットがいる場合の動作を行う。このとき、 $v_{c_1}$  と  $v_{c_2}$  から 2 つのロボットが同時に  $v_{root}$  に移動した場合は、2 台のうち 1 台のロボットが  $pn_i$  を利用して  $v_{c_1}$  または  $v_{c_2}$  に移動する。移動先に既にロボットが配置されていたり、 $v_{root}$  にサイズ 3 のタワーができていたりする場合は、 $pn_i$  を利用して  $mode_i$  を割り当てる。このように、移動先に既にロボットが配置されていたり、 $v_{root}$  にサイズ 3 のタワーができていたりするのは、 $v_{c_1}$  と  $v_{c_2}$  にいるロボットが同時に  $v_{root}$  に移動しようとしたが、片方のロボットだけが先に  $v_{root}$  に移動してしまい、もう片方のロボットが遅れて  $v_{root}$  に移動するために発生する。

#### 4.2 合流フェーズ

タワー生成フェーズの終了時の配置は、根  $v_{root}$  に  $mode_i$  がそれぞれ 0, 1, 2 のロボットが 3 台存在する。合流フェーズでは、まず、センタブロックに含まれていないロボットを移動させる。

次に、サイズ 3 のタワーに含まれるロボットのうち、 $mode_i$  が 1 と 2 である 2 台のロボットと一緒にセンタブロック内のすべての頂点を深さ優先探索する。 $mode_i = 1$  のロボットが  $route_i, exflag_i, vcount_i$  などの変数を使い、合流したロボット数を記録しながら根  $r_{root}$  から深さ優先探索を行う。 $mode_i = 1$  のロボットは、ある頂点に初めて移動したとき、自分が通ってきた親ノードに繋がるポート番号を  $parent$ 、子ノードに繋がるポート番号を  $child_1, child_2$  ( $child_1 > child_2$ ) とすると、 $route_i.push(parent), route_i.push(child_1), route_i.push(child_2)$  の順番で、 $route_i$  にポート番号を入れる。次に、 $route_i.top()$  のポート番号の先の頂点に移動し、 $route_i.pop()$  を行う。このとき、移動先の頂点が子ノード

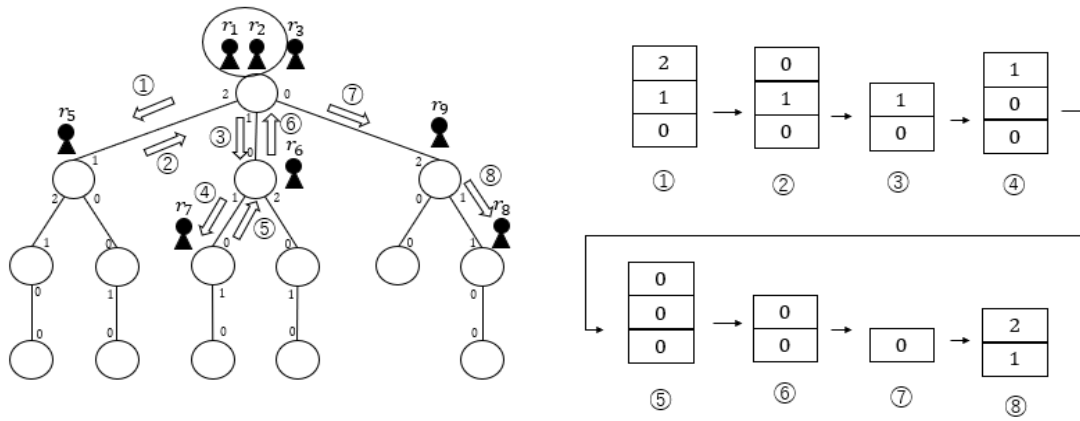


図3  $mode_1 = 1$ であるロボット  $r_1$  が探索したときの  $route_i$  の変化例. 図右側のスタックは,  $route_i$  を表す. 左の図の矢印上にある番号は, 右の図の  $route_i$  の遷移番号に対応している.

なら,  $vcount_i$  をインクリメントする.  $mode_i = 2$  のロボットは,  $mode_i = 1$  のロボットについていくことで探索を行い,  $mode_i = 1$  のロボットが探索が終了し, 別のロボットが  $mode_i = 1$  にして探索を始めるときに目印としての役割を果たす. ロボット  $r_1$  と  $r_2$  の  $mode$  がそれぞれ  $mode_1 = 1$  と  $mode_2 = 2$  のときのロボット  $r_1$  の探索経路と  $route_i$  の変化例を図3に示す.

2台のロボットがセンターブロックサイズと同じ数の頂点を訪問すると, 2台のロボットのうち,  $mode_i = 1$  であるロボットは  $vcount_i$  がセンターブロックサイズと一致するので停止するが, 探索を終えた頂点にいる  $mode_i = 0$  のロボットが  $mode_i$  を1に変更し,  $mode_i = 2$  のロボットと再び探索を行う. ここで, 探索を終えた頂点にいる3台のロボットを  $r_a, r_b, r_c$  とし,  $mode_a = 1, mode_b = 2, mode_c = 0$  とした時,  $mode$  の値が1のロボットが  $r_a$  から  $r_c$  に変更される過程を示す. まず, 探索を終了したことを認識した  $r_a$  は  $mode_a$  を  $-1$  にしながら, 子ノードへ移動する. すると, ロボット  $r_b$  と  $r_c$  によるタワーが作成される. このタワーの子ノードには, ロボットが存在するが, 根  $v_{root}$  以外の場所にタワーができるのは, アルゴリズムの他のフェーズで根に向かうロボットが子のノードから同時に親へ移動してしまった場合だけで, この場合はタワーの子ノードには, ロボットが存在しない. よって, 子のノードにロボットが存在するタワーにいる  $mode_c = 0$  のロボットは, 配置から自身が  $mode_c$  を1に変更しなければならぬと分かるので実際に  $mode_c$  を1に変更しながら親頂点へ進む. そして,  $mode_b = 2$  のロボットが  $mode_c = 1$  のロボットに付いていく形で移動する.  $mode_b = 2$  のロボット  $r_b$  が  $mode_c = 1$  のロボット  $r_c$  に付いていく形で移動した後は,  $mode_a$  を  $-1$  に変更したロボット  $r_a$  はセンターブロックから切り離されていることが分かる. センターブロックから切り離されたロボットは既に他のロボット全て

と合流を行っているので, 再びセンターブロックに戻ってくる必要はない. 図4に  $mode_i = 1$  のロボットが  $r_a$  から  $r_c$  に変更される過程の例を示す. そして,  $mode_i = 1$  のロボットが切り替わると, 新たに  $mode_i = 1$  となったロボットは  $v_{root}$  に進み, そこから再び探索を始める. 最後に  $k - 2$  回, 探索をすると全ロボットは停止し, ペアワイズ合流問題を解くことができる.

**補題 4.1** 空ではない部分木の数が1個で, 多重点が存在しない初期配置から始まる実行は, 有限時間内に根  $v_{root}$  に  $mode_i$  の値がそれぞれ0,1,2の3台のロボットによるタワーが存在する配置に到達する.

**証明.** 初期配置は, 多重点が存在しないことより, ステップ1Aの配置である. ステップ1の配置では,  $v_{root}$  に最も近いロボットがプレイヤーとして選ばれる. 目的地も根  $v_{root}$  方向と変化しない. 多重点を作成されるのは, 同じ親プレイヤーが複数存在し, 同時に根  $v_{root}$  方向に移動したときであり, この移動を行うとフェーズ2の配置に遷移する. ステップ2Aの配置では, サイズ2のタワーに存在するロボットを  $r_i, r_j$  とすると,  $r_i, r_j$  は別々の子ノードから来たロボットなので  $pn_i \not\equiv pn_j \pmod{3}$  が成立するので, 必ず1台のロボットがとどまり, もう1台のロボットが子ノードへ移動し, ステップ1Aの配置に遷移する. よって,  $v_{root}$  にロボットを移動させる過程で, 2台のロボットが同時に親ノードに移動するものが存在しないので,  $v_{root}$  にも1台ずつロボットが移動する. ゆえに,  $v_{root}$  に入った順にロボット1台ずつに  $mode_i$  を割り当てることができる. 以上より, 有限時間内に根  $v_{root}$  に  $mode_i$  の値がそれぞれ0,1,2の3台のロボットによるタワーが存在する配置に到達する. □

部分木が2つや3つの場合も, 1つの場合と同じ方針で, 有限時間内に, 根  $v_{root}$  に  $mode_i$  の値がそれぞれ0,1,2の3台のロボットによるタワーが存在する配置に到達するこ

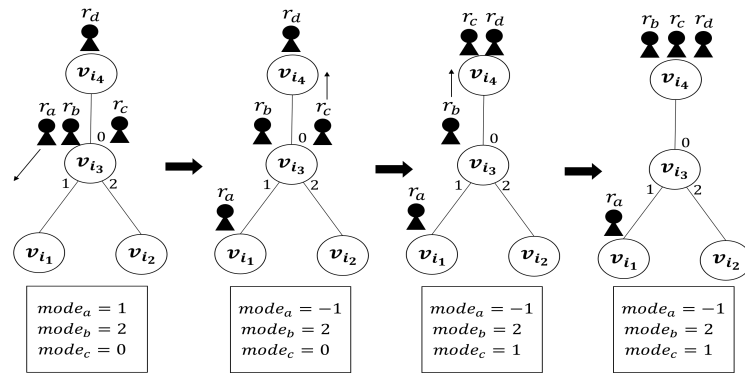


図 4  $mode_i = 1$  のロボットが  $r_a$  から  $r_c$  に変更される過程の例. 図の間の矢印は左のロボットの配置から右のロボットの配置に移したことを表す.

とが示せる.

**補題 4.2** 根  $v_{root}$  に  $mode_i$  の値がそれぞれ 0, 1, 2 の 3 台のロボットによるタワーが存在する配置から始まる実行は, 有限時間にペアワイズ合流問題を解く.

**証明.** 根  $v_{root}$  に  $mode_i$  の値がそれぞれ 0, 1, 2 の 3 台のロボットによるタワーが存在する配置において, センターブロックサイズが  $k-2$  より小さい場合は,  $mode_i = 0$  のロボットが根  $v_{root}$  に移動することでセンターブロックサイズを  $k-2$  にする. この過程で, 同じ親ノードを持つプレイヤーが同時に親頂点に移動してサイズ 2 のタワーが作成された場合は, ステップ 5D の配置となり, タワーのロボットのうち 1 台を子ノードに移動させることでタワーを解消することができる. よって, センターブロックサイズが  $k-2$  になったとき, 根  $v_{root}$  以外には多重点が存在しない. ここで, センターブロックが作成された時に  $mode_i = 2$  のロボットを  $r_a$ ,  $v_{root}$  にいて,  $mode_i = 0$  のロボットを  $r_b$  とする  $mode_i = 1$  のロボットが  $\{r_{c_1}, r_{c_2}, \dots, r_{c_{k-2}}\}$  の順に探索を行ったとする. すると,  $r_{c_i}$  は,  $1 \leq j < i$  の  $r_{c_j}$  と  $r_{c_j}$  の探索によって合流し,  $i < j \leq k-2$  の  $r_{c_j}$  と  $r_{c_i}$  の探索によって合流している.  $r_a$  と  $r_b$  は  $1 \leq i \leq k-2$  の  $r_{c_i}$  と  $r_{c_i}$  の探索によって合流している. 以上より,  $mode_i = 1$  のロボットが  $\{r_{c_1}, r_{c_2}, \dots, r_{c_{k-2}}\}$  の順に探索を行ったとき, 根  $v_{root}$  に  $mode_i$  の値がそれぞれ 0, 1, 2 の 3 台のロボットによるタワーが存在する配置から始まる実行は, 有限時間にペアワイズ合流問題を解く. □

**定理 4.1** 多重点が存在しない初期配置から, 提案アルゴリズムを実行するとペアワイズ合流問題を解くことができる.

**証明.** 補題 4.1 などより, 多重点がない初期配置から根  $v_{root}$  に  $mode_i$  の値がそれぞれ 0, 1, 2 の 3 台のロボットによるタワーが存在する配置に有限時間内に遷移することができる. よって, 補題 4.2 より, 根  $v_{root}$  に  $mode_i$  の値がそれぞれ 0, 1, 2 の 3 台のロボットによるタワーが存在する配置から有限時間内にペアワイズ合流問題を解くことができるので, 多重点がない初期配置から始まる実行は, 提

案アルゴリズムは有限時間内にペアワイズ合流問題を解くことができる. □

## 5. おわりに

本稿では, 中心が 1 つある, または 2 つある場合には 2 つの中心を通る軸に対して非対称な 2 分木で, 初期配置に多重点が存在せず,  $c \geq 3$  の場合で,  $O(\log_2 k)$  ビットの記憶領域を持つロボットによるペアワイズ合流問題を解くアルゴリズムを提案した. 今後は, 本研究で調べたグラフに対して, ペアワイズ合流問題を解くアルゴリズムにおいて,  $c$  の値が実行時間やロボットの性能に与える影響や, 一般的なグラフに対するペアワイズ合流問題を解くアルゴリズムが存在するかを調べる予定である.

## 参考文献

- [1] Paola Flocchini, David Ilcinkas, Andrzej Pelc, and Nicola Santoro, "Computing Without Communicating: Ring Exploration by Asynchronous Oblivious Robots", *Algorithmica*, 65(3), pp. 562–583, 2013.
- [2] Ralf Klasing, Adrian Kosowski, and Alfredo Navarra, "Taking advantage of symmetries: Gathering of many asynchronous oblivious robots on a ring", *Theoretical Computer Science*, 411, pp. 3235–3246, 2010
- [3] Gianlorenzo D'Angelo, Gabriele Di Stefano, Ralf Klasing, and Alfredo Navarra, "Gathering of robots on anonymous grids and trees without multiplicity detection", *Theoretical Computer Science*, 610, pp. 158–168, 2016
- [4] A. Aho, J. Hopcroft, and J. Ullman, "The Design and Analysis of Computer Algorithms." Addison-Wesley Publishing, 1974