法政大学学術機関リポジトリ
HOSEI UNIVERSITY REPOSITORY

PDF issue: 2024-12-21

自律分散ファットロボットに対するグリッド上における集合アルゴリズム

白川, 遥平 / SHIRAKAWA, Yohei

(出版者 / Publisher)
法政大学大学院理工学・工学研究科
(雑誌名 / Journal or Publication Title)
法政大学大学院紀要. 理工学・工学研究科編 / 法政大学大学院紀要. 理工学・工学研究科編
(巻 / Volume)
58
(開始ページ / Start Page)
1
(終了ページ / End Page)
8
(発行年 / Year)
2017-03-31
(URL)
https://doi.org/10.15002/00014225

自律分散ファットロボットに対するグリッド上における集合アルゴリズム

On General Gathering algrithms in Various Grid for Distributed Autonomous Fat Robots

白川 遥平
Yohei Shirakawa
指導教員　和田幸一

法政大学大学院理工学研究科応用情報工学専攻修士課程

This thesis concerns gathering problems for fat robots in seneral grids,such as a two dimensional orthogonal grid, a hexagonal grid and a three dimensional orthogonal grid.We also consider whether robots know a common coordinate system or not.We propose algorithms to solve gathering problems for cases with combinations of the following conditions: schedulers (semi-synchronous or asynchronous), knowledge of the number of robots (known or unknown), radius of robots (small or large) for each grid.

***Key Words** : Autonomous Mobile Robots，Fat Robots，　Gathering Problem*

1.　はじめに

複数の自律的に動作する計算機群が，互いに通信しあいながら共通の目的を持って動作するようなシステムを，分散システムと呼ぶ．その中で自律分散ロボット群とは自律的に動作する複数のロボットが協調的に動作することにより全体でひとつの問題を達成するというものである．この自律分散ロボット群においては全体を効率的に動作させるアルゴリズムの設計が重要である．

自律分散ロボット群の問題のテーマとして，任意の位置に配置されたロボットを一点に集める問題である一点集合問題がある．これまでもロボットの観測や移動の能力や知りうる情報の可解性に関する研究がされている [1-5].

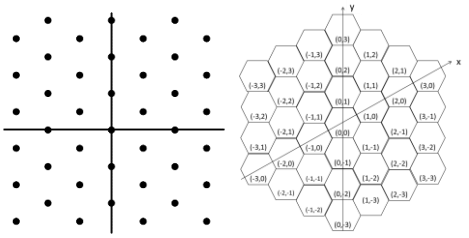
本研究ではロボットが質量を持ち他のロボットの観測および移動において障害物となる可能性を考慮したファットロボットにおいて，六角格子や上や立方格子上を移動する場合の集合問題を考える．文献[6-8]が従来の結果であり，文献[6]では二次元直交座標系における共通座標系を有するファットロボットの集合問題について研究されている.ロボットモデルに関して,格子サイズに対する円盤の半径 2 通り，ロボットの総数に関する知識の有無 2 通り，スケジューラの非同期または半同期の組み合わせによる全 8 通りのモデルについて研究している．文献[7-8]では，同様のモデルに共通の座標系の知識を持たない場合における集合問題を解くアルゴリズムが考えられている．本研究ではその先行研究から平面格子と集合した際の形態が異なる六角格子や立方格子上に拡張した集合問題を解くアルゴリズムを設計する．

2.　モデルの定義

(1)ロボットのモデル

二次元直交座標系 S，(三次元直交座標系 C)は，単位長さ，原点 O，x と y(と z)で識別され互いに直交する 2(3)つの座標軸とその方向，正と負で識別される座標軸の向きで定義される．平面格子 G_S を S のすべての格子点の集合，立方格子 G_C を C のすべての格子点の集合とする．$u=(u_x,u_y)$および $v=(v_x,v_y)$ を G_S 上の点とした時，u と v の距離 $dist_S(u,v)$はマンハッタン距離で定義され $dist(u,v)=|u_x-v_x|+|u_y-v_y|$と定義される．また $u=(u_x,u_y,u_z)$および $v=(v_x,v_y,v_z)$を G_c 上の点とした時，同様にマンハッタン距離で定義され u と v の距離 $dist_S(u,v)$は $dist_c(u,v,z)=|u_x-v_x|+|u_y-v_y|+|u_z-v_z|$となる．

六角格子座標系 H は，(図 1(a))に表される様な $P=\{(\frac{\sqrt{3}}{2}m,n+\frac{1}{2}m)|m,n\in\mathbb{Z}\}$で定義されるの点の集合があり，P に対応する六角格子 $G_H=\{(m,n)|m,n\in\mathbb{Z}\}$ を持ち，単位長さ，原点 O、x 軸と y 軸で識別されお互いに 60°で交わる 2 つの座標軸とその方向，正と負で識別される座標軸の向きで定義されるものである(図 1(b))．　点同士の距離は G_S，G_C 同様にマンハッタン距離で定義される．



(a) (b)

図 1. 六角格子と六角格子座標系

n 台のロボットの集合 R を R={r_0,…,r_{n-1}}とし，ロボットは格子上の点に存在し，格子上の点間を離散的に移動する計算能力を持ったデバイスとする．このロボットは区別不可能で同じアルゴリズムを実行し，匿名で，記憶領域を持たない．また，ロボットは共通の座標系を持つ場合と持たない場合を考える．ロボットが共通の座標系を持つ場合は共通の原点，単位距離，座標軸とその方向および向きを知っており，ロボットはその共通した座標軸において自身の位置の座標を認識できる．また ロボットが共通の座標系を持たない場合は共通の原点，単位距離， 右回り方向のみ知っている，ロボットの視野範囲は自身の点を v とし二次元直交座標系では視野範囲 1 の場合のみを考え，$N_S(v)=\{(u_x,u_y)|(|u_x-v_x|\leqq 1)\wedge(|u_y-v_y|\leqq 1)\}$の集合であり，三次元直交座標系でも視野範囲 1 の場合のみを考え，$N_C(v)=\{(u_x,u_y,u_z)|(|u_x-v_x|\leqq 1)\wedge(|u_y-v_y|\leqq 1)\wedge(|u_z-v_z|\leqq 1)\}$の集合である．六角格子上では視野範囲 1 のみでなく 1 以上の場合も考える．六角格子上で視野範囲 k の時，自身の点を v とし，v からの距離 k となる点の集合 $N_H(v)=\{(u_x,u_y)|dist(v,u)=k\}$とする．視野範囲 1 の際の N_x (v)(x∈{S,C,H})は v にいるロボットが 1 回の移動で移動できる点を表している．ただし，ロボットが同一線上をすれ違う移動と複数のロボットが同一の点に移動することは許されない．また，図 2 に示す様にロボットのサイズでによってはあるロボットは他のロボットが存在する周囲の点へは他のロボットに接触するため移動することができない．ロボットの大きさが十分に小さいロボットが他のロボットが存在しない点に自由に移動できるロボットのサイズを small，周囲のロボットとの接触を避けるため移動に制限が付くロボットのサイズを large と名付ける．平面格子上と六角格子上ではその条件は異なってくる．それぞれの条件を以下に示す．

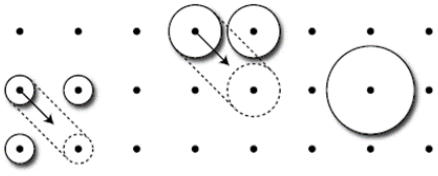


図 2. 平面格子におけるロボットの半径 radius と格子サイズの関係

平面格子上では，ロボットの半径を radius として図 2 の例に示すように，ロボットの大きさが十分に小さく斜めに移動したときに他のロボットに接触することがないモデル small とあるロボットが斜めに移動したとき他のロボットに接触する可能性があるモデル large を

・small:：radius < $\frac{1}{4}\sqrt{2}$

・large： $\frac{1}{4}\sqrt{2} \leqq$ radius < $\frac{1}{2}$

と定義される．

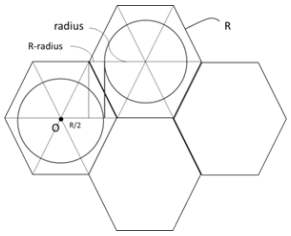


図 3. 六角格子上におけるロボットの半径 radius と格子サイズの関係

図 3 に示す様に六角格子上では平面格子上とは条件が異なる．ロボットの半径の大きさを radius とし radius が線を超えた際にあるロボットが移動した時に周辺のロボットと接触する可能性がある．六角格子上では正六角形の一辺の大きさを R とし

・small：radius < $\frac{3R}{4}$

・large： $\frac{3R}{4} \leqq$ radius < $\frac{\sqrt{3}R}{2}$

と定義する．

立方格子上では平面格子上と同様の$\sqrt{2}$の方向の斜め移動だけでなく$\sqrt{3}$の方向の斜め移動も存在するが，$\sqrt{2}$の方向の斜め移動で接触することがない条件を満たせば，それより幅の広い$\sqrt{3}$の方向の移動でも接触することはない，よって立方格子上での半径 radius の球と変える以外は平面格子上と同様であり

・small:：radius < $\frac{1}{4}\sqrt{2}$

・large： $\frac{1}{4}\sqrt{2} \leqq$ radius < $\frac{1}{2}$

と定義される．

次にロボットの実行モデルについて説明する．各ロボットは以下の 4 つの状態を繰り返し，この 4 つの状態の 1 回の繰り返しをサイクルと呼ぶ．

(1) Wait:ロボットは待機状態．全てのロボットの初期状態はこの状態である．
(2) Look:ロボットは他のロボットの位置を観測する．
(3) Compute:ロボットは観測結果を入力にアルゴ

リズムに従って行き先を計算する．ロボットが無記憶の場合は，入力は直前の Look の情報のみである．

(4) Move:Compute で計算した行き先に移動する

各ロボットが動作する同期の程度によっても図 4 の様な 3 つのモデルが定義される．

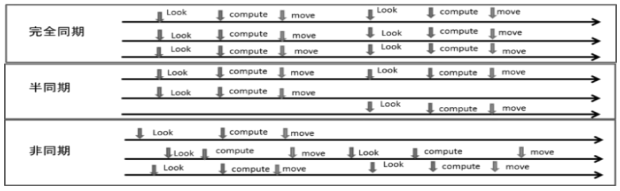


図 4.　同期のモデル

・完全同期(FSYNC):全てのロボットが完全に同期し，全てのロボットが完全に同じ時刻にアルゴリズムを開始し，その後，同じ時刻に Look，Compute，Move を実行する．

・半同期(SSYNC):FSYNC において同じ時刻にのみ Look，Compute，Move が実行されるが，サイクルを実行しないモデルの存在は許すモデルである．

・非同期(ASYNC):FSYNC や SSYNC のようなロボットの実行において同期の仮定を一切置かないモデルである．

これら 3 つのモデルに関して FSYNC のロボットモデルで実現可能な動作は SSYNC のロボットモデルでも実現可能であり，また，SSYNC のロボットモデルで実現可能な動作は ASYNC でも実現可能である．すなわちある問題を ASYNC で解くアルゴリズムが存在する場合は，そのアルゴリズムは SSYNC，FSYNC でも問題を解く．SSYNC と FSYNC の関係も同様である．

動作サイクルの各状態は，仮定するロボットの能力によって細かい定義が異なる．　本研究では，ロボットの視野範囲に制限が存在する．G_x の点 v 上のロボットは Look の結果，ロボットが共通して持つ座標系における $N_x(v)$のうちロボットがいる点の集合 C を得る．Compute で計算される行き先は $N_x(v)$-C のうち現在の座標を除く一点，もしくは移動しないかのいずれかであり，Move における行き先への移動は一瞬であると仮定し移動中のロボットが他のロボットに観測されることはないとする．

Move の移動においてロボットは格子点から格子点へ直接移動する．この時，六角格子上でサイズ Large の場合，基本的にロボットが埋まっている点の隣接点から隣接点への移動はできない．ここで図 5 のような格子の頂点に一度間接的に移動することで，ロボットで埋まっている隣接点から隣接点への移動を可能にする特殊な移動パターンを VI 移動(vertex intermediate move)とする．本研究では，この VI 移動を許すモデルについても考察する．

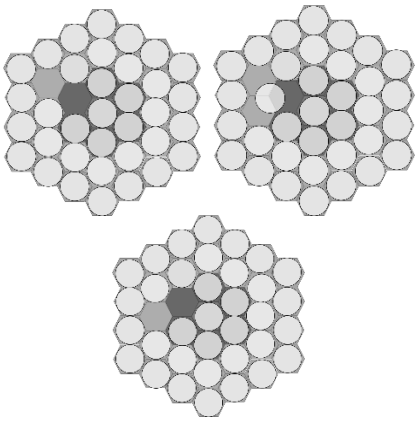


図 5.　VI 移動

⑵集合問題の定義

本研究では，あらかじめ決められた原点 O を中心にロボットを集合させる集合問題を扱う．集合時にロボットが構成する形状が問題の可解性に影響し，原点に近い距離からロボットで埋めていき，集合を達成した際に原点からの距離が最も遠いロボットへの距離が最小となるようにロボットを集合させる．

集合の定義は，グリッド x に対し a_i^x を原点からの距離が i である点の数，原点からの距離が j 以下である点の数を S_j^x とし，ロボットが集合を達成したときの原点から最も遠いロボットまでの距離を $L_{max}{}^x$ とする．原点 O からの距離が i であるロボットの集合を R(i)={r∈R|dist(O，r)=i}としてグリッド上での集合問題を L_{max} を用いて以下のように定義する．

定義　グリッド上で n 台のロボットが次の条件を満たすときロボットは集合したという．

・$\forall r \in R$,　$dist(O, r) \leqq L_{max}{}^x$

・$|R_0|=1$

・$0 < i < L_{max}{}^x$ ，$|R_i| = a_i^x$

従来の研究における平面格子上での a_i^S，S_j^S は以下のようになる

・$a_0^S = 1$

・$a_i^S = 4i\ (i>0)$

・$S_j^S = 2j(j+1)+1\ (j \geqq 0)$

S_j^S の逆関数を考えることで $L_{max}{}^S$ は以下の式で決まる．

$$L_{max}{}^S = \left\lceil \sqrt{\left(\frac{n}{2} - \frac{1}{4}\right)} - \frac{1}{2} \right\rceil$$

本研究で扱う六角格子上での，a_i^H，S_j^H は以下の様になる

・$a_0^H = 1$

・$a_i^H = 6i\ (i>0)$

・$S_j^H = 3j(j+1)+1$ (j≧0)
S_j^Hの逆関数を考えることでL_{max}^Hは以下の式で決まる．
$L_{max}^H=\lceil\sqrt{(\frac{n}{3}-\frac{1}{12})}-\frac{1}{2}\rceil$

また立方格子上では
・$a_0^C = 1$
・$a_i^C = 4i^2 + 2$ (i>0)
・$S_j^C = \frac{4j^3}{3}+2j^2+\frac{8j}{3}+1$
となり，L_{max}^CはS_j^Cとロボットの台数 n から
$\frac{4j^3}{3}+2j^2+\frac{8j}{3}+1-n=0$ を満たす実数解の小数点以下の切り上げで得られる．このグリッド x 上の集合問題を解くアルゴリズムの集合を，同期のモデル sync∈{async,ssync,fsync}，ロボットの n を知るか否か num∈{known,unknown}，ロボットの大きさのモデル size ∈ {small,large} をパラメータとし，GA^x(sync,num,size)で定義する．

(3)緒定義
六角格子上での右回りの表記
二次元直交座標系 S において u=(u_x,u_y)，v=(v_x,v_y)の 2 点を考えた時，(u_x*v_y - u_y*v_x)<0 を満たすとき原点を中心に v は u の右回りに位置するという[6]. 六角格子座標系 H においても，格子点上の 2 点，u=(u_x,u_y)，v=(v_x,v_y)を考えた時，$u'=(u_x'=\frac{\sqrt{3}}{2}u_x,\quad u_y'=u_x+\frac{1}{2}u_y)$，$v'=(v_x'=\frac{\sqrt{3}}{2}v_x,\quad v_y'=v_x+\frac{1}{2}v_y)$として二次元直行座標系の値に適用し，($u_x$'*$v_y$'-$u_y$'*$v_x$')<0 を満たすとき原点を中心に v'は u'の右回りに位置するとする．
ある v'∈H に対して，N_H(v)の点のうち下記の条件を満たす点に対して表記を定義する．(図 6)
p_v':(dist(O,p_v')=dist(O,v')-1)∧(v_x'p_y'-v_y'p_x'<0)
e_v':(dist(O,e_v')=dist(O,v'))∧(v_x'e_y'-v_y'e_x'<0)
s_v':(dist(O,s_v')=dist(O,v')+1)∧(v_x's_y'-v_y's_x'≦0)
さらにそれと逆方向の移動を
Lp_v':(dist(O,p_v')=dist(O,v')-1)∧(v_x'p_y'-v_y'p_x'>0)
Le_v':(dist(O,e_v')=dist(O,v'))∧(v_x'e_y'-v_y'e_x'>0)
Ls_v':(dist(O,s_v')=dist(O,v')+1)∧(v_x's_y'-v_y's_x'≧0)

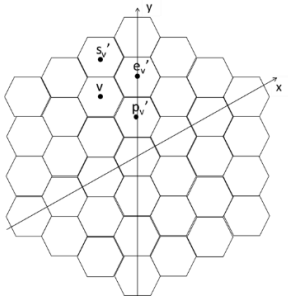


図 6. 点 v とその周囲の点の例

3. 従来の結果と提案アルゴリズム
先行研究では平面格子状でのアルゴリズムが提案されており，7 通りに対して集合問題を解くアルゴリズムを考え，残り 1 つのモデルに対して，その非可解性について述べられている．その結果が表 1 である．本研究では平面格子と集合した際の形態が異なる六角格子や立方格子上に拡張した集合問題を解くアルゴリズムを設計する．その結果が表 2，3 である．

表 1. 平面格子上でのアルゴリズム

二次元直交座標			small	large
共通座標系有	async	unknown	Spiral	非可解
		known	(Spiral)	SpiralSpread
	ssync	unknown	(Spiral)	PullSlide
		known	(Spiral)	(SpiralSpread) (PullSlide)
共通座標系無	async	unknown	未解決	非可解
		known	未解決	未解決
	ssync	unknown	未解決	未解決
		known	GridPullSpin2	GridPullSpin3

表 2. 六角格子上でのアルゴリズム

六角格子座標			small	large
共通座標系有	async	unknown	hexagonSpiral	非可解
		Known	(hexagonSpiral)	(非可解※1※2)
	ssync	unknown	(hexagonSpiral)	(非可解※1※2)
		known	(hexagonSpiral)	非可解※1※2 VIMalgorithm※3
共通座標系無	async	unknown	未解決	(非可解)
		known	未解決	(非可解※1※2)
	ssync	unknown	未解決	(非可解※1※2)
		known	hexagonPullSpin	(非可解※1※2)

表 3. 立方格子上でのアルゴリズム

三次元直交座標 共通座標系有		small	Large
async	unknown	Spiral3	未解決
	known	(Spiral3)	未解決
ssync	unknown	(Spiral3)	未解決
	known	(Spiral3)	未解決

(1) サイズ Small の場合
共通座標系知識有の場合については，平面格子上では(async,unknown,small)モデルについてのアルゴリズムが提案されていた．そのアイデアを座標系によらない形に一般化し，それを六角格子上に適用することで六角格子上の共通知識有の場合についてのアルゴリズム hexagonSpiral ∈ GA^H(async,unknown,small)を設計した．
平面格子上でのアルゴリズムが Spiral である．

アルゴリズム 1 点 $v = (v_x, v_y)$ 上のロボット r 上のアルゴリズム *Spiral*

1: **Predicate**:
2: $Spin(r) \equiv (v \neq 1 \vee y > 0) \wedge (e_v \notin C)$
3: $Upper(r) \equiv (v = 1) \wedge (y \leq 0) \wedge ((v_x - 1, v_y) \notin C)$
4: $Stay(r) \equiv (v = O) \vee (\neg Spin(r) \wedge \neg Upper(r))$
5: **Actions**:
6: Spin :: $Spin(r)$ → 行き先を e_v に
7: Upper :: $Upper(r)$ → 行き先を $(v_x - 1, v_y)$ に
8: Stay :: $Stay(r)$ → 移動しない

アルゴリズムの概要

非同期に移動する複数のロボットがすれ違うまたは同一の点に移動してしまうことがないように，ロボットの通り道を一本道に設計することで集合を達成するアルゴリズムである．

平面格子状でのアルゴリズム Spiral のアイデアを一般化アルゴリズムが GeneralAlgorithm1 であり，これによって定理 1 を示す．

ハミルトン閉路の順序付け(next(v))は右回りの方向($\vec{v}$)で行う
原点からの各距離毎に1点ずつ存在する
o_iの集合Oとe_iの集合Eを以下のように再帰的に定義する．

(i)原点をe_0と定め,原点からの距離1の点から任意でe_1を1つ定める
$e_0e_1 \in E$
(ii)If $\overrightarrow{eiv}$ ∧ N(e_i) ∧ dist(e_i)=dist(v)
o_i=v∈O
If $\overleftarrow{eiv}$ ∧ N(e_i) ∧ dist(e_i)=dist(v)+1
o_{i+1}=v∈O
If $\overrightarrow{oiv}$ ∧ N(o_i) ∧ dist(o_i)=dist(v)
e_i=v∈E

GeneralAlgorithm1
座標系上の原点からの距離iの点v=(vx,vy)のロボットrのアルゴリズム
If （v∈O ∧ (移動先にロボットが存在しない)）
→e_{i-1}に移動する
Else if （v∉O ∧ (移動先にロボットが存在しない)）
→同距離内で$\vec{v}$の方向に移動する
Else 移動しない

アルゴリズムの概要

格子平面状において，各距離ごとに距離に対する入り口の点と出口の点を集合によって定め，その入り口から出口ハミルトン経路を作成することで，平面格子上に一本道を作成する．

定理 1　GeneralAlgorithm1 により一般の平面格子上に一方通行の経路を作成することができる．

このアルゴリズムを六角格子上に適用したアルゴリズムが hexagonSpiral である．

定理 2　hexagonSpiral∈GA^H(async,unknown,small)

アルゴリズム hexagonSpiral

1: Predicate:
2: Up(r) ≡ (vx≦0 ∧ (vy=-1)) ∧ (p_v'∉C)
3: Spin(r) ≡ ¬Up(r) ∧ (e_v'∉C)
4: Stay(r) ≡ (v=O) ∨ (¬Up(r) ∧ ¬Spin(r))
5: Actions:
6: Up::Up(r) → 行き先をp_v'へ
7: Spin::Spin(r) → 行き先をe_v'へ
8: Stay::Stay(r) → 移動しない

(2) 共通の座標系知識を持たずサイズ Small の場合

共通座標系知識無の場合については，平面格子上では(async,unknown,small)モデルについてのアルゴリズム pullspin が提案されていた．そのアイデアを六角格子上に適用したものがアルゴリズム hexagonPullSpon である．

Algorithm 1 点 $v = (v_x, v_y)$ 上のロボット r のアルゴリズム

1: **Predicate:**
2: $PullBlocked(r) \equiv p_v \in \mathcal{C}$
3: $SpinBlocked(r) \equiv s_v \in \mathcal{C}$
4: $SpinStop(r) \equiv e_v \in \mathcal{C}$
5: $Onthe(r) \equiv (v_x = 0) \vee (v_y = 0)$
6: $StopDistance(r) \equiv dist(O, v) = L_{max}$
7: $GatheringStop(r) \equiv StopDistance(r) \wedge ((dist(O, v) = 1) \vee \neg OntheAxes(r))$
8: $Pull(r) \equiv \neg PullBlocked(r) \wedge \neg OntheAxes(r)$
9: $Spin(r) \equiv \neg Pull(r) \wedge \neg SpinBlocked(r) \wedge \neg SpinStop(r) \wedge \neg GatheringStop(r)$
10: $Stay(r) \equiv \neg Pull(r) \wedge \neg Spin(r)$
11: **Actions:**
12: Pull::$Pull(r)$ → 行き先を p_v に
13: Spin::$Spin(r)$ → 行き先を s_v に
14: Stay::$Stay(r)$ → 移動しない

アルゴリズムの概要

原点に対し，右回りで距離を縮める pull の移動と原点に対し右回りで同距離の点に移動する spin の移動を繰り返すことで集合を達成する．

定理 3　hexagonPullSpin は六角格子上において共通の座標系を持たず(ssync,known,small)モデルにおいての集合問題解く．

アルゴリズム hexagonPullSpin

1: Predicate:
2: PullBlocked(r) ≡ p_v'∈C
3: SpinBlocked(r) ≡ e_v'∈C
4: Spinstop(r) ≡ s_v'∈C
5: ontheAxes(r) ≡ (vx=0) ∨ (vy=0) ∨ (|vx|=|vy|)
6: StopDistance(r) ≡ dist(O,v) = L_{max}
7: GatheringStop(r) ≡ StopDistance(r) ∧ ((dist(O,v)=1) ∨ ¬ontheAxes(r)
8: Pull(r) ≡ ¬PullBlocked(r) ∧ ¬ontheAxes(r)
9: Spin(r) ≡ ¬Pull(r) ∧ ¬SpinBlocked(r) ∧ ¬Spinstop(r) ∧ ¬GatheringStop(r)
10; Stay(r) ≡ ¬Pull(r) ∧ ¬Spin(r)
11: Actions:
12: Pull::Pull(r) → 行き先をp_v'へ
13: Spin::Spin(r) → 行き先をe_v'へ
14: Stay::Stay(r) → 移動しない

(3)ロボットサイズ Large の場合

六角格子上においてロボットサイズ large で視野範囲 1 の時，平面格子上における視野範囲 1 の時に見えていた距離 2 の斜めの位置のロボットを見ることができない．また移動制限も平面格子上に比べ厳しくなっている．これにより平面格子状では可解であったモデルでも非可解となっており，その非可解性について示す．また，移動の制限を緩和するため

に特殊な移動パターンを許可した場合について集合問題を解くアルゴリズムを示す.

定理 3 サイズ Large かつ視野範囲 1 のロボットで集合問題を解くアルゴリズムは存在しない

証明の概要
平面格子上においてサイズ Large の場合において集合問題を解くアルゴリズムでは距離が 2 である斜めの位置のロボットが見えておりその情報を利用したアルゴリズムを設計していた. しかし六角格子上でサイズ Large の場合では, その範囲が見えていないので同じ条件では集合をとくことができないことを示した.

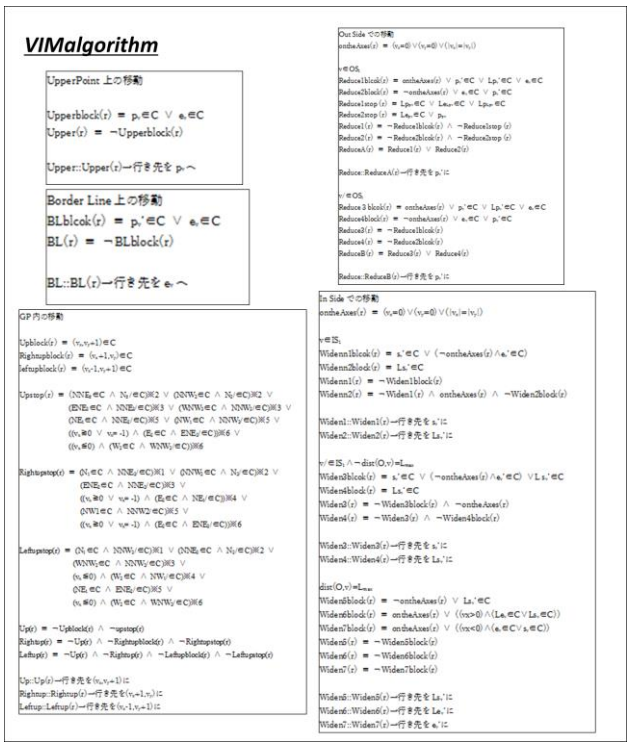
定理 3 から視野範囲 1 の場合においては全ての場合で集合問題と解くことができないことを示した. これによりサイズ Large のモデルでは視野範囲 2 以上の場合も考察し定理 4 を示した.

定理 4 サイズ large かつ視野範囲 2 以上のロボットで初期状態に制限のない場合に一般には集合問題を解くアルゴリズムは存在しない.

証明の概要
特定のロボットの台数 n と特定の初期状態において集合達成不可能である構成があることを示し, 一般に集合を達成するアルゴリズムが存在しないことを示した.

サイズ Large のモデルにおいて非可解性を示したことから, これに当てはまらないモデルの初期状態に制限があり,視野範囲 2 で VI 移動有の場合に集合問題を解くアルゴリズムを設計した. そのアルゴリズムが VIMalgorithm である.

定理 6 VIMalgorithm$\in GA^{H}$(ssync,known,large)※
※VI 移動あり, 初期状態制限有, 視野範囲 2



アルゴリズムの概要
n を元に集合を達成する構成である. GP を計算する. そして GP の内部ではロボットが奥から順に詰まるように移動していく. GP の外ではデットロックが起こらない距離までロボットが広がる移動をする In Side と, デットロックが起こる可能性がない距離では距離を縮める移動をする Out Side, またその境界の Border Line と Upper Point の 5 つのエリアによってロボットが移動を変える戦略である.
このアルゴリズムによって定理 6 を示す.

(3)立方格子上のアルゴリズム
立方格子上で(ssync,known,large)のモデルでは, 平面格子と六角格子で利用した汎用アルゴリズムの戦略を三次元上で適用することで同様のモデルにおいて集合問題を解くアルゴリズムを設計した. そのアルゴリズムが Spiral3 である.
アルゴリズム上で使用する移動パターンとして平面格子上では $u=(u_x,u_y)$, $v=(v_x,v_y)$を二次元直交座標系 S 上の 2 点とし, $(u_xv_y-u_yv_x)<0$ を満たす時, 原点を中心に v は u の右回りに位置するとしており,
・e_v:$(dist(O,p_v)=dist(O,v)-1)\wedge (v_xp_y-v_yp_x\leqq 0)$ が存在した.
ここでは右回りの移動を考える際三次元上で 2 軸を固定し, 平面上での右回りを考える.
$u=(u_x,u_y,u_z)$, $v=(v_x,v_y,v_z)$を三次元直交座標系 C 上の 2 点とした時, xz 平面上での右回りの移動を
・e_{vxz}: $(dist(O,e_{vxz})=dist(O,v)-1)\wedge (e_{vpz}-e_{ypx}<0)$とし他の

平面も同様の定義とする.
さらにそれと逆方向の移動を
・Le_{vxz}: $(dist(O,e_{vxz})=dist(O,v)-1)\wedge(v_{xez}-v_{yex}\geqq 0)$ とし他の平面も同様の定義とする.

Spiral3

```
//If v=o_i
 If (vx=0∧vy=0)∧dist(O,r)=odd∧(vz>0)∧((vx,vy,vz-1)∉C)
  →(vx,vy,vz-1)
 If
(vx=0∧vy=0)∧dist(O,r)=even∧(vz<0)∧((vx,vy,vz+1)∉C)
  →(vx,vy,vz+1)

//Else v=¬o_i
 //If  v=o_i'
  If (vz>0){
   If (vz=odd){
    If (dist(O,r)=even)∧(vy=0∧vx<0)∧(pyz∉C)
     →pyz
    If (dist(O,r)=odd)∧(vy=1∧vx>0)∧(Lpyz∉C)
     →Lpyz }
   If (vz=even){
    If (dist(O,r)=even)∧(vy=0∧vx≦0)∧(pyz∉C)
     →pyz
    If (dist(O,r)=odd)∧(vy=1∧vx≧0)∧(Lpyz∉C)
   →Lpyz }
  }
  If (vz<0){
   If (vz=odd){
    If (dist(O,r)=even)∧(vy=-1∧vx≦0)∧(Lpyz∉C)
     →Lpyz
    If (dist(O,r)=odd)∧(vy=0∧vx≧0)∧(pyz∉C)
     →pyz }
   If (vz=even){
    If (dist(O,r)=even)∧(vy=-1∧vx<0)∧(Lpyz∉C)
     →Lpyz
    If (dist(O,r)=odd)∧(vy=0∧vx>0)∧(pyz∉C)
     →pyz }
 }
 If vz=0{
   If (dist(O,r)=even)∧(vy=0∧vx<0)∧(Lpxz ∉C)
    →Lpxz
   If (dist(O,r)=odd)∧(vy=0∧vx>0)∧(Lpxz∉C)
    →Lpxz
 }
 //Else If v=¬o_i'
  Else (pxy∉C)
  →pxy
```

アルゴリズムの概要

汎用アルゴリズムの戦略を拡張し，三次元としての距離を縮める点がその点の存在する平面へ近づけるために同距離の二次元平面上に各 1 ずつ距離を縮める点を設計することで一方通行の経路を作成した.

定理 7　Spiral3∈GA^C(async,unknown,small)

4.　結果と今後の課題

本研究では, ロボットサイズが小さい時に一般の座標系上集合問題を解くアルゴリズムと, それを六角格子上と立方格子上に適用したアルゴリズムを提案した. またロボットサイズが大きい時には, 六角格子上で視野範囲 1 では非可解であり, 2 以上でも一般では非可解であることを示した. そして特殊な移動を許した際に初期状態に制限があり視野範囲2の時, 六角格子上集合問題を解くアルゴリズムを示した. 今後の研究課題としては六角格子座標系上で特殊な移動を許した際の視野範囲 1 での非可解性は考察や三次元直交座標系上での他のモデルにおける集合問題を解くアルゴリズムの考察があげられる.

謝辞

本研究を行うにあたり, 多大なご指導を賜りました和田幸一教授, ならびに名古屋工業大学院工学研究科情報工学専攻片山喜章教授に深く感謝致します. また日常的に議論を通じて多くの知識や示唆を頂いた和田研究室の皆様に感謝致します.

参考文献

[1] Bandettini, A.,Luporini, F. and Viglietta, G.: A Survey on Open Problems for Mobile Robots, ArXiv e-prints (2011).

[2] Cieliebak, M., Flocchini, P., Prencipe, G. and Santoro, N.: Solving the Robots Gathering Problem, Automata, Languages and Programming (Baeten, J., Lenstra, J., Parrow, J. and Woeginger, G., eds.), Lecture Notes in Computer Science, Vol. 2719, Springer Berlin Heidelberg, pp. 1181–1196 (online), DOI: 10.1007/3-540-45061-0 90 (2003).

[3] X. Defago, M. Gradinariu, S. Messika and P. Raipin-Parvedy, Fault-tolerant and self-stabilizing mobile robots gathering, Proc. of 20th International Symposium on Distributed Computing, pp.46-60, 2006.

[4] Flocchini, P., Prencipe, G., Santoro, N. and Widmayer, P.: Gathering of asynchronous robots with limited visibility, Theoretical Computer Science, Vol. 337, No. 1-3, pp. 147 – 168 (online), DOI: 10.1016/j.tcs.2005.01.001 (2005).

[5] Prencipe, G.: On the Feasibility of Gathering by Autonomous Mobile Robots, Structural Information and Communication Complexity (Pelc, A. and Raynal, M., eds.), Lecture Notes in Computer Science, Vol. 3499, Springer Berlin Heidelberg, pp. 246–261 (online), DOI: 10.1007/11429647 20 (2005).

[6] 伊藤公一, 片山喜章,和田幸一 : ”共通座標系を有するファットロボットのグリッド上での集合について”, 第*143* 回アルゴリズム研究会, 研究報告アルゴリズム*(AL),2013-AL-143(2),1-8(2013-02-22).*

[7] 伊藤佳進, 片山喜章,和田幸一:”共通の座標系を有しないグリッド平面上におけるファットロボッ

トの集合”,研究報告アルゴリズム(AL),
2014-AL-148(9),1-7(2014-06-06).
[8] 伊藤佳進, 片山喜章,和田幸一:”グリッド上でのLarge サイズファットロボットの集合について”,コンピュテーション研究会 講演論文集,pp.1-8,Mar.2015.
[9]長尾英剛：三次元グリッド空間における自律分散ロボットの集合に関する研究,名古屋工業大学情報工学科論文,2016-02.
[10] 長尾秀剛, 片山喜章, 金鎔煥, 和田幸一, 三次元グリッド空間における自律分散ロボットの集合の研究, 電子情報通信学会総合大会,
D-22-2,2016-03.

本論文に関する発表

[1] 自律分散ファットロボットの様々なグリッド上での集合問題について,電子情報通信学会技術研究報告 = IEICE technical report : 信学技報
[2] 自律分散ファットロボットに対する様々なグリッド上における汎用集合アルゴリズムについて,第12 回情報科学ワークショップ,2016