

# 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

伊藤 公一<sup>1,a)</sup> 片山 喜章<sup>1,b)</sup> 和田 幸一<sup>2,c)</sup>

**概要:** 本稿では、格子平面上において共通座標系を有し円盤で表される自律分散ロボットの集合問題を扱う。ロボットモデルに関して、格子の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 (2 通り) の組み合わせによる全 8 通りのモデルのうち 7 通りに対して集合問題を解くアルゴリズムを与える。残り 1 つのモデルに対してその非可解性を示す。

**キーワード:** 自律分散ロボット, ファットロボット, 集合問題

## On Gathering Problems in Grid for Fat Robots with the Common Coordinate System

**Abstract:** This paper studies gathering problems in grid for Fat robots. We assume robots have the common coordinate system. We propose algorithms to solve gathering problems for the seven cases of eight models defined by combinations of the scheduler (async. or semi-sync.), knowledge about the number of robots (unknown and known), and radius of fat robots (large or small). We also show that the last case of async., unknown and large is unsolvable.

**Keywords:** autonomous mobile robots, fat robots, gathering problem

### 1. はじめに

近年、自律分散システムに関する研究が盛んにおこなわれており、自律分散ロボット群の研究もそのひとつである。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の集合であり、これら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目的を達成する。深海や宇宙など人間が直接管理することが難しいような環境において、人間による管理の必要がない自律分散ロボット群の利用が期待されている。

本研究では、自律分散ロボット群を計算論的な観点からとらえ、その協調問題を扱う。ロボットは他のロボット

の位置を観測 (*Look*) し、観測結果を入力にアルゴリズムに従って行き先を計算 (*Compute*) し、その行き先に移動 (*Move*) するというシンプルなサイクルを繰り返し問題を解く。よく扱われる問題に一点集合問題がある。一点集合問題とは、任意の位置に配置されたロボットがあらかじめ決められていない一点に集合する問題であり、ロボットの観測や移動の能力や知りうる情報と問題の可解性に関して研究されている [1-4]。

これらの研究では、ロボットは平面上を自由に移動する点で表されている。実際のロボットを考えたとき、ロボットは物理的な大きさを持ち、ロボットが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すべきである。これに対し、ロボットを平面上の点ではなく円盤で表すファットロボット (*fat robots*) が考えられており、ファットロボットの集合問題を扱った研究がされている [5-8]。

文献 [1] は、ファットロボットの集合問題を理論的に扱った最初の研究である。文献 [6] では、連続平面と離散平面におけるファットロボットの集合問題について研究されて

<sup>1</sup> 名古屋工業大学大学院工学研究科情報工学専攻  
Nagoya Institute of Technology, Graduate School of Computer Science and Engineering

<sup>2</sup> 法政大学理工学部応用情報工学科  
Hosei University, Faculty of Science and Engineering Department of Applied Informatics

<sup>a)</sup> kito@moss.elcom.nitech.ac.jp

<sup>b)</sup> katayama@nitech.ac.jp

<sup>c)</sup> wada@hosei.ac.jp

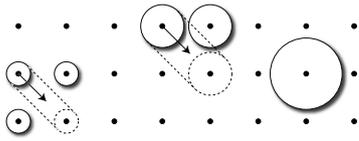


図1 ロボットの半径  $radius$  と格子サイズの関係を表す例．左からそれぞれ， $radius < \frac{1}{4}\sqrt{2}$ ， $\frac{1}{4}\sqrt{2} \leq radius < \frac{1}{2}$ ， $\frac{1}{2} \leq radius$ ．

表1 モデルと提案アルゴリズム．

		<i>small</i>	<i>large</i>
非同期	$n$ を知らない	<i>Spiral</i>	非可解
	$n$ を知っている	<i>Spiral</i>	<i>SpiralSpread</i>
半同期	$n$ を知らない	<i>Spiral</i>	<i>PullSlide</i>
	$n$ を知っている	<i>Spiral</i>	<i>SpiralSpread</i> <i>PullSlide</i>

おり，離散平面におけるアルゴリズムでは，複数のロボットが同時に同一の点に移動してしまう問題をロボットが直接通信を行い一方のロボットを優先させるという手法で解決している．また，ロボットが他のロボットの観測において障害物となることがないモデルでの集合問題についても研究されている [7]．文献 [8] では，視野範囲に制限のあ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．

本稿では，格子平面上において共通座標系を有し円盤で表される自律分散ロボットの集合問題を扱う．格子上のファットロボットの集合に関しては [6] で研究されているが，複数のロボットが同時に同一の点に移動してしまう問題を本稿ではロボットどうしの通信を行わずに解決する．さらに本稿では，ロボットの半径  $radius$  によって次の2つのモデルを定義する．

- *small*:  $radius < \frac{1}{4}\sqrt{2}$
- *large*:  $\frac{1}{4}\sqrt{2} \leq radius < \frac{1}{2}$

モデル *small* では，図1に例を示すように，ロボットの大きさが十分に小さく斜めに移動したとき他のロボットに接触することはない．一方でモデル *large* では，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り，ロボットどうしの接触を避けるようにアルゴリズムを設計する必要がある．本稿では  $radius \geq \frac{1}{2}$  であるロボットに関しては考慮しない．ロボットの半径について2通り，ロボットがロボットの台数を知りうるかどうか，サイクルの実行が非同期または半同期かのそれぞれ2通りの組み合わせ全8通りのモデルのうち，7つのモデルにおいて集合問題を解く3つのアルゴリズム *Spiral*，*SpiralSpread*，*PullSlide* を提案する．また，残りの1つのモデルにおいてその非可解性を示す．モデルと提案内容の対応表を表1に示す．

## 2. モデルと問題定義

### 2.1 ロボットモデル

二次元直交座標系  $\mathcal{F}$  は，単位長さ，原点  $O$ ， $x$  と  $y$  で識別され互いに直交する2つの座標軸とその方向 (*direction*)，正と負で識別される座標軸の向き (*orientations*) で定義される．格子  $\mathcal{G}$  を  $\mathcal{F}$  のすべての整数座標とする． $u = (u_x, u_y)$  および  $v = (v_x, v_y)$  を  $\mathcal{G}$  上の点としたとき， $u$  と  $v$  の距離  $dist(u, v)$  を  $dist(u, v) = |u_x - v_x| + |u_y - v_y|$  とする．

$n$  台のロボットの集合  $\mathcal{R}$  を  $\mathcal{R} = \{r_0, r_1, \dots, r_{n-1}\}$  とする．ロボットは  $\mathcal{G}$  上の点に存在し， $\mathcal{G}$  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である．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と仮定する．つまり，ロボットは共通の原点，単位距離，座標軸とその方向および向きを知っており，ロボットはその座標系において自分の座標を認識することができる． $v = (v_x, v_y)$  を  $\mathcal{G}$  上の点としたとき， $v$  と  $v$  の周囲8点の集合を  $N(v) = \{(u_x, u_y) \mid (|u_x - v_x| \leq 1) \wedge (|u_y - v_y| \leq 1)\}$  とする． $v$  上のロボットの視野範囲を  $N(v)$  とし，一回で移動できる点を  $N(v)$  に含まれる点とする．ただし，ロボットが同一直線上をすれ違う移動と，複数のロボットが同時に同一の点へ移動してしまうことは許されない．さらに，ロボットサイズ *large* では， $x$  座標および  $y$  座標をとともに増減させる移動 (斜め移動) を行うロボットが他のロボットに接触することは許されない．

以下，多くの研究で一般に用いられている標準的なロボットの実行モデルについて説明する．各ロボットの動作は次の状態の繰り返しであり，これらの状態の一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．ロボットは無制限に待機することはできない．開始時点ですべてのロボットが Wait 状態である．
- (2) **Look**: ロボットはセンサから他のロボットの位置座標を得る．Look の結果，ロボットは他のロボットの位置の集合  $C$  を得る．
- (3) **Compute**: ロボットはアルゴリズムに従って行き先を計算する．ロボットは無記憶であるため，アルゴリズムの入力は直前の Look で得た  $C$  のみである．
- (4) **Move**: Compute 状態において計算した行き先に向かって移動する．

各ロボットが実行する動作の同期の程度によって次の3つのモデルが定義される．

- 完全同期 (**fully-synchronous**, **FSYNC**): すべてのロボットが完全に同期した時計を用いてアルゴリズムを実行できると仮定する．すなわち，すべてのロボットは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 のそれぞれを実行する。

- 半同期 (semi-synchronous, SSYNC): FSYNC において、サイクルを実行しないロボットの存在を許すモデルである。ただし、すべてのロボットは無限回サイクルを実行すると仮定する。
- 非同期 (asynchronous, ASYNC): ロボット実行に関する (上記のような) 仮定を一切置かないモデルである。各ロボットのサイクル開始時刻や Look, Compute, Move にかかる時間の上限も与えられない。

これらの3つの同期モデルに関して、SSYNC は FSYNC の実行を実現可能であり、ASYNC は SSYNC, FSYNC の実行を実現可能である。つまり、ある問題を ASYNC で解くアルゴリズムが存在する場合、そのアルゴリズムは SSYNC, FSYNC でも正しく問題を解く。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。本稿では、ロボットの視野範囲に制限があり、ロボットが共通の座標系知っていると仮定している。 $\mathcal{G}$  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  $N(v)$  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  $N(v)$  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。

ここまでロボットの透明および不透明については述べていないが、本稿でのモデルではロボットの視野範囲の関係上ロボットは透明でも不透明でもかまわない。視野範囲の広さ、ロボットの透明性によって問題の可解性が変化する可能性がある。他にも、座標系についてどこまで合意があるかによっても可解性が変化する可能性がある。

## 2.2 集合問題の定義

本稿では、あらかじめ決められた一点 (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するが、本研究では、原点に近い点からロボットで埋められていて、ロボットが集合したとき原点から最も遠いロボットまでの距離が最小になるようにする。このとき、図 2(a) の例に示すように、ロボットが集合したときの形はひし形になり、図 2(b) のように正方形に集まっているときロボットは集合していない。

$a_i$  を原点からの距離が  $i$  である点の数とし、原点からの距離が  $j$  以下である点の数を  $S_j$  とすると、 $a_i$  および  $S_j$  は以下ようになる。

$$\begin{cases} a_0 = 1 \\ a_i = 4i \quad (i > 0) \end{cases}, \quad S_j = 2j(j+1) + 1 \quad (j \geq 0)$$

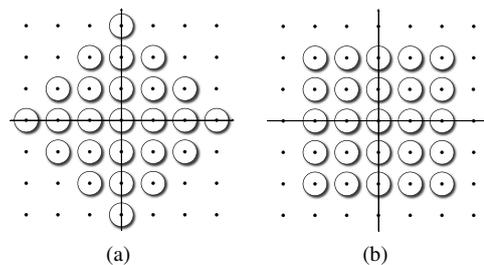


図2 ロボットが集合している例 (a) と、集合していない例 (b)

ロボットが集合を達成したとき、原点から最も遠いロボットまでの距離を  $\rho$  とすると、 $\rho$  は  $S_k \geq n$  を満たす最小の  $k$  となる。 $S_j$  の逆関数を考えることで、 $\rho$  は以下の式で求まる。

$$\rho = \left\lceil \sqrt{\frac{n}{2} - \frac{1}{4}} - \frac{1}{2} \right\rceil \quad (n \geq 1)$$

原点  $O$  からの距離が  $i$  であるロボットの集合を  $R_i = \{r \in \mathcal{R} | \text{dist}(O, r) = i\}$  とし、本稿で扱う集合問題を  $\rho$  を用いて以下のように定義する。

定義 2.1 (集合問題).  $n$  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。

- $\forall r \in \mathcal{R}, \text{dist}(O, r) \leq \rho$
- $\rho > 1 \rightarrow \forall i, 0 \leq i < \rho, |R_i| = a_i$

集合問題とは、任意の位置に配置された  $n$  台のロボットを有限時間内に集合させ、集合したのちすべてのロボットを停止させる問題である。□

ロボットが集合したとき、 $|R_\rho|$  はロボットの総数  $n$  によって変化し、この数は以下の式で表され、以降  $\lambda$  と呼ぶ。

$$\lambda = n - S_{\rho-1} = n - 2\rho(\rho-1) - 1$$

$\lambda < 4\rho$  であるとき、原点からの距離が  $\rho$  である点のうちロボットが停止する点は任意とする。

本稿では、同期モデル  $\text{sync} \in \{\text{async}, \text{ssync}, \text{fsync}\}$ 、ロボットが  $n$  を知るか否か  $\text{num} \in \{\text{known}, \text{unknown}\}$ 、ロボットの大きさのモデル  $\text{size} \in \{\text{small}, \text{large}\}$  をパラメータとし、集合問題を解くアルゴリズムのクラスを  $\mathcal{GA}(\text{sync}, \text{num}, \text{size})$  で定義する。

## 2.3 諸定義

$u = (u_x, u_y), v = (v_x, v_y)$  を  $\mathcal{F}$  上の2点としたとき、 $(u_x v_y - u_y v_x) < 0$  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ひ、 $u$  は  $v$  の左回りに位置するという。また、 $(u_x v_y - u_y v_x) = 0$  を満たすとき、 $u, v$  および原点  $O$  が同一直線上に位置するという。

ある  $v \in \mathcal{G}$  に対して、 $N(v)$  の点のうちいくつかの点について以下の通り名前を付ける (図 3 参照)。

- $p_v : (\text{dist}(O, p_v) = \text{dist}(O, v) - 1) \wedge (v_x p_y - v_y p_x \leq 0)$
- $e_v : (\text{dist}(O, e_v) = \text{dist}(O, v)) \wedge (v_x e_y - v_y e_x < 0)$
- $s_v : (\text{dist}(O, s_v) = \text{dist}(O, v) + 1) \wedge (v_x s_y - v_y s_x < 0)$

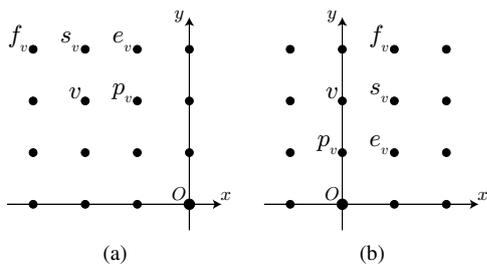


図3 点  $v$  とその周囲の  $p_v, s_v, e_v, f_v$  の例.

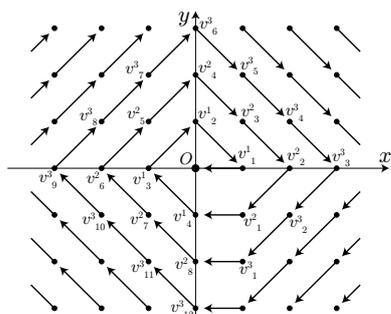


図4 アルゴリズム *Spiral* の移動パターンおよび  $ES_1 = \langle v^1_1, \dots, v^1_4 \rangle$ ,  $ES_2 = \langle v^2_1, \dots, v^2_8 \rangle$ ,  $ES_3 = \langle v^3_1, \dots, v^3_{12} \rangle$  を示す.

- $f_v$ :  $(dist(O, f_v) = dist(O, v) + 2)$ . 複数候補がある場合、原点を中心に右回りの点.

## 2.4 アルゴリズムの記述

本稿では、アルゴリズムの実行を以下の形式（ガード付きアクション）で表す.

$\langle label \rangle :: \langle guard \rangle \rightarrow \langle action \rangle$

各アクションはラベル付けされており、各アクションのガード  $\langle guard \rangle$  は Look の結果からなる論理式で表される. ロボットはガードが真の場合のみ命令文  $\langle action \rangle$  を実行する.

## 3. アルゴリズム *Spiral*

アルゴリズム *Spiral*  $\in \mathcal{GA}(async, unknown, small)$  について説明する.

ロボットサイズが *small* のとき、斜めに移動するロボットが他のロボットに接触することはない. しかし、非同期に移動するロボットがすれ違う移動をすることなく、また複数のロボットが同一の点に移動してしまうことなく集合を達成しなくてはならない. このため、アルゴリズム *Spiral* ではロボットの通り道を一本道する. ロボットは図4の実線矢印で示す順路を移動していく. このとき、ロボットは移動先にロボットがいなくて移動する. 一本道なので、複数のロボットが同一の点へ移動したり、すれ違うように移動してしまうことはない.

ロボット  $v$  は原点からの距離が原点から  $v$  までの距離と等しく、原点を中心に右回りの点 ( $e_v$ ) に移動する. こ

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

ここで、原点からの距離が  $i$  であり  $x$  座標が 1 である点を  $v^i_1 = (1, -i + 1)$  としたとき、 $v^i_1$  から開始して原点からの距離が  $i$  である点を左回りに順番に辿った系列を次のように定義する.

定義 3.1 (Equidistance Sequence).  $i > 0$  を任意の整数、 $v^i_1 = (1, -i + 1)$  とする.  $\mathcal{G}$  の相異なる点からなる系列  $\langle v^i_1, \dots, v^i_{4i} \rangle$  が各  $j(1 \leq j < 4i)$  に対して  $v^i_j = e_{v^i_{j+1}}$  を満たすとき、この系列を  $ES_i$  という. □

図4に  $ES_1, ES_2, ES_3$  を示す.  $ES_i$  を  $\langle v^i_1, \dots, v^i_{j-1}, v^i_j, \dots, v^i_{4i} \rangle$  としたとき、 $v^i_j$  上のロボットは Look の結果に  $v^i_{j-1}$  が含まれないとき  $v^i_{j-1}$  に移動する. また、 $ES_{i-1}$  を  $\langle v^{i-1}_1, \dots, v^{i-1}_{4(i-1)} \rangle$  としたとき、 $v^i_1$  上のロボットは Look の結果に  $v^{i-1}_{4(i-1)}$  が含まれないとき  $v^{i-1}_{4(i-1)}$  に移動する.

以上の戦略を実現したアルゴリズムをアルゴリズム 1 に示す.

ロボットが *Spiral* に従い移動すると、原点からの距離が  $\rho - 1$  以下である点と、 $ES_\rho$  を  $\langle v^\rho_1, \dots, v^\rho_{4\rho} \rangle$  としたときの  $v^\rho_1$  から  $v^\rho_\lambda$  までの点にすべてのロボットが移動し、以降移動を行うロボットは存在せず、このときロボットは集合を達成している. 後述のアルゴリズム *SpiralSpread*, *PullSlide* においても同じ配置にロボットを集合させるため、以降この点の集合を  $GP$  と呼ぶ. また、集合を達成するまでの間少なくとも 1 台のロボットは移動可能であり、複数のロボットが同時に同一の点に存在することはない. 以下の定理が成り立つ.

定理 3.1.  $Spiral \in \mathcal{GA}(async, unknown, small)$

## 4. アルゴリズム *SpiralSpread*

アルゴリズム *SpiralSpread*  $\in \mathcal{GA}(async, known, large)$  について説明する.

ロボットサイズ *large* における斜めに移動するロボットと他のロボットとの接触を避けるため、*SpiralSpread* では斜めの移動を行わずにロボットを集合させる. アルゴリズムは、 $n$  をもとに  $GP$  を求め  $GP$  以外の点から  $GP$  へ移動できる点を一点に限定し、その点に向かって  $GP$  の回りを螺旋上に  $GP$  に近づいていく戦略と、 $GP$  の内部でロボットが奥から順に詰まるように移動していく戦略で構成され

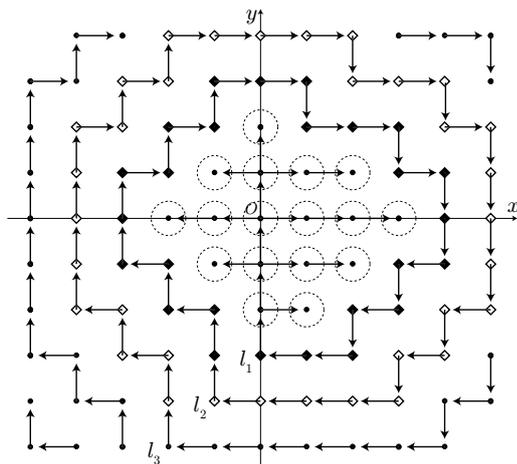


図5  $n = 17$  のときのアルゴリズム *SpiralSpread* の移動パターン (実線矢印). 破線で囲われた点が集合を達成したときロボットがいる点 ( $GP$ ) である.

る. 図5中の実線矢印によってロボットの移動パターンを示す.  $GP$  へ螺旋上に近づいていく経路は一本道である.  $GP$  の内部での移動は,  $y$  軸上を入り口から奥の方向へ移動する移動と,  $x$  軸方向に広がる移動を行う. ここは一本道ではないが, ロボットがすれ違う移動および複数のロボットが同時に同一の点に移動してしまふことはない.

ここで,  $GP$  に向かう螺旋状の経路を定義するため,  $GP$  を覆う一つ外側の層  $L_1$  (図5の  $\blacklozenge$ ), さらにその外側の層  $L_2$  (図5の  $\diamond$ ), 以下同様に  $L_i$  ( $i > 0$ ) を次の通り再帰的に定義する.

$$\begin{cases} L_0 = GP \\ L_{i+1} = \{v \in \mathcal{G} \mid ((N(v) \cap L_i) \neq \emptyset) \wedge (v \notin \bigcup_{k=0}^i L_k)\} \end{cases}$$

$L_i$  の点上のロボットは同じ  $L_i$  の点のうち原点を中心に右回りの点へ移動する.

$L_i$  から  $L_{i-1}$  に移動する点  $l_i$  は,  $L_i$  に含まれ  $x$  座標が  $1-i$ ,  $y$  座標が  $0$  以下である点とする. 図5に  $l_1, l_2, l_3$  の例を示す.

以上の戦略を実現したアルゴリズムをアルゴリズム2に示す. ロボットが *SpiralSpread* に従い移動したとき, すべてのロボットが  $GP$  に移動し集合するまでの間少なくとも1台のロボットは移動可能であり, 複数のロボットが同時に同一の点に存在することはない. 以下の定理が成り立つ.  
定理 4.1.  $SpiralSpread \in \mathcal{GA}(async, known, large)$

## 5. アルゴリズム *PullSlide*

アルゴリズム *PullSlide*  $\in \mathcal{GA}(ssync, unknown, large)$  の説明およびその正当性の証明を行う.

### 5.1 アルゴリズムの説明

アルゴリズム *PullSlide* では, ロボットは斜めの移動をせず,  $x$  座標または  $y$  座標を1増減させる移動のみを行う. ロボットは原点  $O$  からの距離を1小さくする移動 *Pull* と距離を1大きくする移動 *Slide* の2通りの移動で  $O$  の方へ

## アルゴリズム2 点 $v = (v_x, v_y)$ 上のロボット $r$ 上のアルゴリズム *SpiralSpread*

- 1: **variable:**
- 2:  $\forall v \notin GP$  であるとき  $v \in L_i$  であるとし, 以下の変数を用いる.
- 3:  $spin$ :  $L_i \cap N(v)$  に含まれ, 原点を中心に  $v$  より右の点
- 4:  $upper$ :  $L_{i-1} \cap N(v)$  に含まれ,  $v$  からの距離が1である点
- 5: **Predicate:**
- 6:  $Spin(r) \equiv (v \neq l_i) \wedge (next \notin C)$
- 7:  $Upper(r) \equiv (v = l_i) \wedge (upper \notin C)$
- 8:  $Above(r) \equiv (v \in GP) \wedge (v_x = 0) \wedge ((v_x, v_y + 1) \in GP) \wedge ((v_x, v_y + 1) \notin C)$
- 9:  $Right(r) \equiv (v \in GP) \wedge (v_x \geq 0) \wedge ((v_x + 1, v_y) \in GP) \wedge ((v_x + 1, v_y) \notin C)$
- 10:  $Left(r) \equiv (v \in GP) \wedge (v_x \leq 0) \wedge ((v_x - 1, v_y) \in GP) \wedge ((v_x - 1, v_y) \notin C)$
- 11: **Actions:**
- 12:  $Spin$ :  $Spin(r) \rightarrow$  行き先を  $spin$  に
- 13:  $Upper$ :  $Upper(r) \rightarrow$  行き先を  $upper$  に
- 14:  $Above$ :  $Above(r) \rightarrow$  行き先を  $(v_x, v_y + 1)$  に
- 15:  $Right$ :  $\neg Above(r) \wedge Right(r) \rightarrow$  行き先を  $(v_x + 1, v_y)$  に
- 16:  $Left$ :  $\neg Above(r) \wedge \neg Right(r) \wedge Left(r) \rightarrow$  行き先を  $(v_x - 1, v_y)$  に
- 17:  $Stay$ :  $\neg Spin(r) \wedge \neg Upper(r) \wedge \neg Above(r) \wedge \neg right(r) \wedge \neg Left(r) \rightarrow$  移動しない

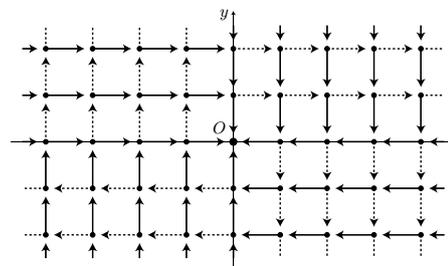


図6 Pullでの移動先(実線)およびSlideの移動先(破線)

移動していく. ただし,  $v$  上のロボットは, Pullでは原点からの距離を1小さくする点のうち  $p_v$  に, Slideでは  $s_v$  に移動する(図3参照). 図6に各点におけるPullおよびSlideの移動先を示す. ロボット  $v$  は  $p_v$  にロボットが存在しない場合 Pullを行い,  $p_v$  にロボットが存在しかつ  $s_v$  にロボットが存在しない場合 Slideを行う.

次に, 複数のロボットが同時に同一の点へ移動してしまふことを防ぐ戦略について述べる. 図6から複数のロボットが同時に同一の点へ移動してしまふパターンは, (A) 原点へ移動するロボットどうしの衝突, (B) SlideするロボットとPullするロボットの衝突, (C) Pullするロボットどうしの衝突の3パターンであることがわかる. (A)の場合, 原点にPullできるロボットを座標  $(1, 0)$  に存在するロボットに限定し, その他の点  $(0, 1), (-1, 0), (0, -1)$  のロボットはSlideにより原点回りを移動することで衝突を避ける. (B)の場合, Pullするロボットを優先させ, Slideするロボットを停止させる. つまり点  $v$  上のロボットは, 原点からの距離が自分より2大きく, 原点を中心に自分より右回りの点  $f_v$  にロボットが存在するときSlideを行わない. (C)はPullの移動先が  $x$  軸または  $y$  軸上となる場合である. このとき,  $x$  軸または  $y$  軸上を移動するロボットを優先させる. つまり  $v$  上のロボットは, 原点からの距離が自分と等しく,

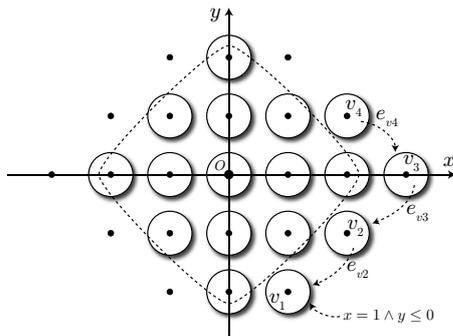


図7 ロボットを停止させる戦略の例．破線枠内のロボットは移動しない． $x = 1$  で  $y \leq 0$  の点は Slide を行わない．それ以外のロボットは破線矢印の指す点にロボットがいるとき Slide を行わない．

原点を中心に右回りの点  $e_v$  に注目する． $e_v$  上のロボット存在し，かつ  $e_v$  が  $x$  軸または  $y$  軸上であるとき， $v$  上のロボットは Pull を行わない．

ここまで，ロボットの移動の戦略と，移動先の重複を避ける戦略について述べた．しかし，現在の戦略のみでは，永遠に移動し続けるロボットが存在し集合が達成できない．原点に近い点からロボットで埋めていき，そのロボットを停止させたい．原点からの距離が  $i$  以下である点すべてにロボットが存在する状況を考える．原点からの距離が  $i+1$  である点にロボットが到達したとしても，そのロボットは再び Slide で離れてしまう可能性がある．そこで，系列  $ES_{i+1}$  を  $\langle v_1^{i+1}, \dots, v_{4i}^{i+1} \rangle$  としたとき， $v_1^{i+1}$  に到達したロボットから停止させ（Slide を行わないようにし）， $v_2^{i+1}, v_3^{i+1}$  と順に到達したロボットから停止させたい．まず， $v_1^{i+1}$  つまり  $x$  座標が 1， $y$  座標が 0 以下である点のロボットは Slide を行わない．次に， $v_2^{i+1}$  上のロボットは  $v_1^{i+1}$  つまり  $e_{v_2^{i+1}}$  にロボットがいるとき Slide を行わない．この繰り返しにより  $ES_{i+1}$  にロボットを停止させていく．図7の例では， $v_1$  は  $x = 1$  かつ  $y \leq 0$  を満たすため移動しない．また， $v_2$  から  $v_4$  もそれぞれ  $e_{v_2}$  から  $e_{v_4}$ （破線矢印の指す先）にロボットがいるため Slide を行わない．このとき，原点からの距離が  $i$  以下でロボットいない点が存在するとき，原点からの距離が  $i+1$  以上の点のロボットのうち少なくとも1つがやがて Pull 可能となり，ロボットすべてが永久に停止してしまうことはない．

以上の戦略を実現したアルゴリズムがアルゴリズム3である．

## 5.2 正当性の証明

ロボットが Pull および Slide で移動するとき，複数のロボットが同時に同一の点へ移動してしまうパターンは5.1節で説明した通りであり，アルゴリズムではそのいずれに対してもロボットの衝突を避けている．

ここで，ある  $v \in \mathcal{G}$  について， $v$  および Pull の繰り返し

## アルゴリズム3 点 $v(v_x, v_y)$ 上のロボット $r$ 上のアルゴリズム

### $\Delta$ PullSlide

- 1: **Predicate:**
- 2:  $OriginBlocked(r) \equiv (dist(O, v) = 1) \wedge (v_x \neq 1)$
- 3:  $SlideBlocked(r) \equiv f_v \in C$
- 4:  $PullBlocked(r) \equiv (e_v \in C) \wedge (e_v \text{ が } x \text{ 軸または } y \text{ 軸上})$
- 5:  $OuterBorder(r) \equiv e_v \in C$
- 6:  $StopPoint(r) \equiv (v_x = 1) \wedge (v_y \leq 0)$
- 7:  $Pull(r) \equiv (v \neq O) \wedge (p_v \notin C) \wedge \neg OriginBlocked(r) \wedge \neg PullBlocked(r)$
- 8:  $Slide(r) \equiv (v \neq O) \wedge \neg Pull(r) \wedge (s_v \notin C) \wedge \neg SlideBlocked(r) \wedge \neg OuterBorder(r) \wedge \neg StopPoint(r)$
- 9:  $Stay(r) \equiv \neg Pull(r) \wedge \neg Slide(r)$
- 10: **Actions:**
- 11:  $Pull::Pull(r) \rightarrow$  行き先を  $p_v$  に
- 12:  $Slide::Slide(r) \rightarrow$  行き先を  $s_v$  に
- 13:  $Stay::Stay(r) \rightarrow$  移動しない

のみで  $v$  に到達できる点からなる集合  $PCS(v)$  を考える． $PCS(v)$  は次のように再帰的に定義できる．

(1)  $\forall v \in \mathcal{G}, v \in PCS(v)$

(2)  $v \in PCS(v)$  かつ  $p_u = v$  ならば， $u \in PCS(v)$

補題 5.1.  $v \in \mathcal{G}$  を原点でない任意の点とする．ある時刻  $t$  において， $v$  にロボットが存在せず， $PCS(v)$  に少なくとも1台のロボットが存在するとする． $v$  上にロボットが存在する時刻  $t' (> t)$  が存在する．

略証． $t$  以降の任意の時刻において， $PCS(v)$  に含まれる点に存在するロボットのうち， $v$  からの距離が最小であるロボットの集合を考える．その集合に含まれるロボットの少なくとも1つは Pull でき，Pull を行ったロボットは  $v$  までの距離を1小さくする（11行目）．これを繰り返すことで，やがて有限時間内に題意が成り立つことが証明できる．□

補題 5.2. 有限時間内に原点にロボットが到達し，それ以降移動しない．

略証． $v_1 = (1, 0)$ ， $v_2 = (0, 1)$ ， $v_3 = (-1, 0)$ ， $v_4 = (0, -1)$  とする．アルゴリズムより， $v_1$  に存在するロボットは原点に移動する（11行目）．一方， $v_2, v_3, v_4$  に存在するロボットは Pull は行わず，Slide によってそれぞれ  $(1, 1)$ ， $(-1, 1)$ ， $(-1, -1)$  に移動する（12行目）．これを繰り返すことで， $v_1, \dots, v_4$  のうちどれかにロボットが存在すると，やがてロボットは原点に到達する．また補題 5.1 より，有限時間内に  $v_2, v_3, v_4$  にロボットが到達する．以上より，題意が成り立つ．□

ここで，ロボット全体が集合する前，原点に近い点から順にロボットで埋まっていて局所的にロボットが集合している部分を考える．

定義 5.1 (Sub-Gathering Border, Sub-Gathering Set). ある時刻  $t$  において，以下を満たす最大の整数  $i$  を  $SGB(t)$  とする．

$$|R_0| = 1, |R_1| = 4, |R_2| = 8, \dots, |R_i| = 4i$$

また，原点からの距離が  $SGB(t)$  以下であるロボットの集合を  $SGS(t)$  とする．□

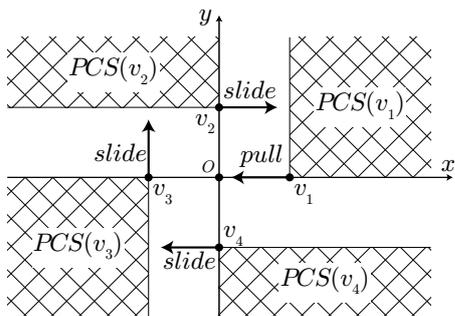


図8  $PCS(v_1), \dots, PCS(v_4)$  はそれぞれ網掛け部分に含まれる点になる。

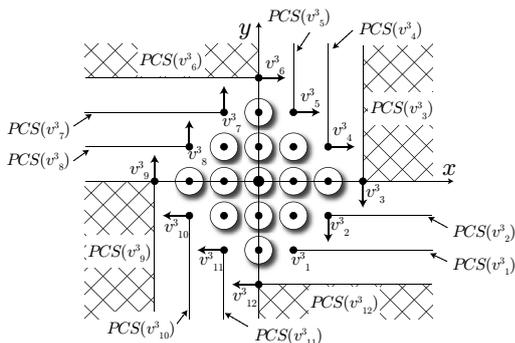


図9  $SGB(t) = 2$  のときの  $PCS$  および  $ES_3 = \langle v_1^3, \dots, v_{12}^3 \rangle$ . 軸上の点に対する  $PCS$  は網掛けの部分に含まれる点であり、そうでない点に対する  $PCS$  は実線上の点である。また、各  $v_2^3, \dots, v_{12}^3$  における Slide の移動先を実線矢印で示す。

補題 5.3. 任意の時刻  $t$  において、 $SGS(t)$  に含まれるロボットは移動しない。

証明.  $SGB(t) = 0$  のとき、原点のロボットは移動しない。原点からの距離が  $SGB(t)$  であるロボットは Pull を行うことはできず、述語 *OuterBorder* が成り立つため Slide も行うことはできない。 □

補題 5.4. 任意の時刻  $t$  に対し、 $SGB(t) \leq \rho - 2$  ならば、 $SGB(t') = SGB(t) + 1$  を満たす  $t' (> t)$  が存在する。

略証.  $i = SGB(t) + 1$  とするとき、有限時間内に  $ES_i = \langle v_1^i, \dots, v_{4i}^i \rangle$  の各点にロボットが順に到達し、到達したロボットは以降移動しないことを帰納法により示す。

まず、 $v_1^i$  にロボットが到達することを示す。図9に、 $PCS(v_1^i), \dots, PCS(v_{4i}^i)$  および  $v_2^i$  から  $v_{4i}^i$  上のロボットの Slide による移動先の例を示す。 $j$  を  $2 < j \leq 4i$  を満たす整数とすると、図9から分かるように、 $v_j^i$  上のロボットの Slide による移動先は  $PCS(v_{j-1}^i)$  となる。さらに、 $PCS(v_1^i) \cup \dots \cup PCS(v_{4i}^i) = \mathcal{G} \setminus SGB(t)$  となることが分かる。 $v_1^i$  にロボットが到達しないと仮定すると、 $\mathcal{G} \setminus SGB(t)$  に含まれる点上にロボットが存在しないことになり矛盾する。 $v_1^i$  に到達したロボットは、述語 *StopPoint* が成り立つため、以降移動することはない。

次に、各  $v_m^i (1 < m \leq n)$  にロボットが到達し、そのロボットが以降移動しないと仮定する。このとき、有限時間内に  $v_{n+1}^i$  にロボットが到達し、以降そのロボットが移動しない

ことを示す。 $v_1^i$  にロボットが到達することと同様の戦略で  $v_{n+1}^i$  にロボットが到達することを示すことができる。 $v_{n+1}^i$  に到達したロボットは述語 *OuterBorder* が成り立つため、以降移動することはない。 □

補題 5.4 より、 $SGB(t) = \rho - 1$  を満たす  $t$  が存在することがわかる。このとき、 $\mathcal{G} \setminus SGS(t)$  であるロボットは原点からの距離が  $\rho$  である点で停止しなくてはならない。 $ES_\rho = \langle v_1^\rho, \dots, v_{4\rho}^\rho \rangle$  としたとき、 $v_1^\rho$  から  $v_\lambda^\rho$  までのすべて点にロボットが到達しそれらのロボットが以降移動しなくなる時刻  $t' (> t)$  が存在することは、補題 5.4 の証明と同じ議論によって示すことができる。以上より以下の定理が成り立つ。

定理 5.1.  $PullSlide \in \mathcal{GA}(ssync, unknown, large)$  □

## 6. 非可解性

ここでは、 $\mathcal{GA}(async, unknown, large)$  に属するアルゴリズムが存在しないことを示す。このモデルにおいて集合問題を解くアルゴリズム  $\mathcal{A}$  が存在すると仮定し、 $\mathcal{A}$  に従うロボットの移動を表す静的なグラフ（移動グラフ）を次のように定義し、移動グラフが満たす性質から  $\mathcal{A}$  が存在しないことを示す。

定義 6.1 (移動グラフ). 移動グラフ  $M\mathcal{G}_{\mathcal{A}}(V: \text{頂点}, E: \text{有向辺})$  は以下のように定義される。

- $V = \mathcal{G}$
- $\langle u, v \rangle \in E$ : 頂点  $u$  に存在するロボットが  $\mathcal{A}$  に従って  $v$  に移動する可能性がある。

□

以降、グラフ理論における用語を使用する。また、ここまで使用してきた二次元平面における用語をグラフにおいても同様に使用する。

補題 6.1.  $M\mathcal{G}_{\mathcal{A}}$  に入次数が 2 以上である頂点は存在しない。

証明. ロボットが 2 台の場合を考える。ある  $i, j, k \in V$  について、 $(\langle i, k \rangle \in E) \wedge (\langle j, k \rangle \in E)$  が成り立つと仮定する。 $j \in N(i)$  が成り立つとき  $i \in N(j)$  が成立し、 $i, j$  上のロボットは互いに存在を知ることができる。一般性を失うことなく、 $i$  上のロボットより  $j$  上のロボットを優先させると仮定する。しかし、非同期スケジューラでは Look したときの状況と Move するときの状況が異なる可能性があり、 $i$  および  $j$  上のロボットが同時に  $k$  に移動してしまうスケジュールが存在する。例えば、 $i$  上のロボットが  $j$  にロボットがいないことを観測し  $k$  への移動を決定したとする。このとき、 $i$  が Look を終えた時刻から Move するまでの間に  $j$  にロボットが到達し、さらにそのロボットが  $k$  への移動を行ってしまう可能性がある。以上より、題意が成り立つ。 □

補題 6.2. 入辺のみである頂点をシンクと呼び、出辺のみである頂点をソースと呼ぶ。 $\mathcal{G}_{\mathcal{A}}$  に 2 つ以上のシンクが存在することはなく、シンクが存在する場合それは原点である。また、 $\mathcal{G}_{\mathcal{A}}$  にソースは存在しない。

証明．最終的にロボットが集合を達成したときにロボットが存在する点以外にシンクがあった場合，初期状況においてその点にロボットが存在すると集合は達成されない．ロボットの台数がわからない場合，原点以外にシンクが存在してはならない．

シンク同様，最終的にロボットが集合したときにロボットが存在する点にソースがある場合，初期状況においてその点にロボットが存在しないと集合が達成できなくなる．したがって，ソースは存在しない． □

補題 6.3. 任意の  $v \in V$  から原点までの有向経路が存在する．証明．集合の定義より原点に必ずロボットが到達しなければならないことは明らか． □

補題 6.4.  $MG_{\mathcal{A}}$  は有向無閉路グラフになる．

証明． $MG_{\mathcal{A}}$  に有向閉路が含まれると仮定する．このとき，補題 6.1 より有向閉路が単純閉路であることがいえる．また， $MG_{\mathcal{A}}$  に 2 つ以上の閉路が含まれるとすると，入次数が 2 以上となる点が必ず存在するため，2 つ以上の閉路が含まれることはない．このとき， $MG_{\mathcal{A}}$  全体が一つの閉路となる場合と， $MG_{\mathcal{A}}$  は 1 つの閉路と，閉路のうち一つの頂点から原点までの有向経路で構成される場合が考えられる．

閉路に含まれる点が有限個である場合を考える．このとき，閉路に含まれる点がロボットが集合する点に含まれるとき，初期状況で閉路以外の部分にすべてのロボットがいると，集合することができない．

次に閉路に含まれる点が無限個である場合を考える．このとき，ロボットは無限長の閉路を移動することになり有限時間内に集合できない． □

補題 6.5.  $MG_{\mathcal{A}}$  に斜め移動に相当する有向辺は存在しない．証明．一般性を失わず， $v_1 = (x, y)$ ， $v_2 = (x + 1, y)$ ， $v_3 = (x, y + 1)$ ， $v_4 = (x + 1, y + 1)$  とし， $\langle v_1, v_4 \rangle \in E$  であるとする．ロボットサイズ  $large$  では， $v_2$  または  $v_3$  上にロボットが存在するとき， $v_1$  から  $v_4$  にロボットが移動することは許されない． $v_2$ ， $v_3$  への有向辺について考える．補題 6.1 と同様に考えると， $v_1$ ， $v_4$  以外の点から  $v_2$  または  $v_3$  への有向辺はあってはならない．

$v_2$  または  $v_3$  がソースとならないためには， $v_1$  または  $v_4$  から  $v_2, v_3$  への有向辺が存在する必要がある．このとき， $v_1$  または  $v_4$  の出次数が 3，もしくは  $v_1$  および  $v_4$  の出次数が 2 となる．このとき，どの場合においても， $MG_{\mathcal{A}}$  に閉路ができるか，シンクの数か 2 以上となるか，入次数が 2 以上となる点が存在することになり，補題 6.1，6.2，6.4 に矛盾する． □

定理 6.1.  $\mathcal{GA}(async, unknown, large)$  に属するアルゴリズムは存在しない．

証明．ロボットが 3 台のときを考える．補題 6.2 より，原点が唯一のシンクとなる．補題 6.1 および補題 6.5 より，原点への有向辺が存在する頂点は原点からの距離が 1 であ

る点のいずれか 1 頂点であり，この頂点を  $v_1$  とする．同様に， $v_1$  への有向辺が存在する頂点は原点からの距離が 2 である点のいずれか 1 つであり，この頂点を  $v_2$  とする．初期状況において 3 台のロボットが原点， $v_1$ ， $v_2$  に存在したとき，すべてのロボットが移動できず，集合できない．

## 7. おわりに

本稿では，共通座標系を持つファットロボットらの格子上で集合問題について考えた．ロボットの同期性，ロボット総数  $n$  の知識の有無，ロボットのサイズが異なるモデルにおいてそれぞれ問題の可解性・非可解性を示した．今後の課題には，座標系に関する知識の有無，ロボットの視野範囲の変更，ロボットが透明かどうかなどロボットのモデルを変更したときに問題の可解性についての考察がある．

## 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] 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).
- [3] 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).
- [4] 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).
- [5] Czyzowicz, J., Gąsieniec, L. and Pelc, A.: Gathering few fat mobile robots in the plane, *Theoretical Computer Science*, Vol. 410, No. 6-7, pp. 481–499 (online), DOI: 10.1016/j.tcs.2008.10.005 (2009).
- [6] Cord-Landwehr, A., Degener, B., Fischer, M., Hullmann, M., Kempkes, B., Klaas, A., Kling, P., Kurras, S., Martens, M., Heide, F., Raupach, C., Swierkot, K., Warner, D., Weddemann, C. and Wonisch, D.: Collisionless Gathering of Robots with an Extent, *SOFSEM 2011: Theory and Practice of Computer Science* (Černá, I., Gyimóthy, T., Hromkovič, J., Jefeřery, K., Kráľovič, R., Vukolić, M. and Wolf, S., eds.), Lecture Notes in Computer Science, Vol. 6543, Springer Berlin Heidelberg, pp. 178–189 (online), DOI: 10.1007/978-3-642-18381-2\_15 (2011).
- [7] Gan Chaudhuri, S. and Mukhopadhyaya, K.: Gathering Asynchronous Transparent Fat Robots, *Distributed Computing and Internet Technology* (Janowski, T. and Mohanty, H., eds.), Lecture Notes in Computer Science, Vol. 5966, Springer Berlin Heidelberg, pp. 170–175 (online), DOI: 10.1007/978-3-642-11659-9\_17 (2010).
- [8] Bolla, K., Kovacs, T. and Fazekas, G.: Gathering of Fat Robots with Limited Visibility and without Global Navigation, *Swarm and Evolutionary Computation* (Rutkowski, L., Korytkowski, M., Scherer, R., Tadeusiewicz, R., Zadeh, L. and Zurada, J., eds.), Lecture Notes in Computer Science, Vol. 7269, Springer Berlin Heidelberg, pp. 30–38 (online), DOI: 10.1007/978-3-642-29353-5\_4 (2012).