

進化型確率的有限状態機械を用いたスワームロボットの集合行動生成

片田 喜章 (摂南大学)

Aggregation of Swarm Robots Using Evolutionary Probabilistic Finite State Machine

Yoshiaki KATADA (Setsunan University)

Abstract : This paper proposes to use evolutionary computations to design probabilistic finite state machine for the controller on an aggregation problem of swarm robotics. This problem formulated as an optimization problem was solved by the PSOs. Several computer simulations were conducted to investigate the validity of the proposed method. The results obtained in this paper show to us that the proposed method is useful for the aggregation problem and the best optimized controllers are interpretable and scalable. This would be transferable to real swarm robots problems.

1. 緒言

スワームロボティクス (SR) とよばれる分野 [1][2] が近年、注目を集めている。その中でも Sahin ら [3] は単一ロボットでは問題解決が困難もしくは非効率的なタスクに対して、構造が簡単で安価に構築でき制御器を含めて完全に均質な自律ロボット (シミュレーションではなく実機) を多数 (最低でも 10-20 台) 用いて解決策を提示しようというアプローチを提案している。上述の設計理念に従うと、各ロボットの構成は制御器を含めて均質であることが望ましい。均質でありながら、個々のロボットが多様な、時として役割分担的な振舞いを示すことが要求される。このように設計された SR は頑健性・拡張性・柔軟性という特性を示すことが期待される [1]。

SR では、これまでに空間的な組織化行動・探索行動・集団的意思決定といったタスクが取り扱われてきた [1]。空間的な組織化行動とは、集合・パターン形成および構造物の自己組立や形成を指す。空間において、どのようにロボットと物体を組織化し分散させるか (配置するか)、どのように凝集したロボットが構造物を形成するかについて焦点を当てた研究が行われている。群れに関する大域的な情報の取得と相互通信が可能でかつ精度が求められる場合には、中央集権的な制御手法が適用される。精度は保証されるが、頑健性・拡張性・コストの面で問題がある。一方、群れに関する局所的な情報取得と近隣程度のロボット同士の通信が可能である場合、分散的な制御手法が適用される。精度は期待できないものの、頑健性・拡張性・構築の容易さといった利点がある [4]。本研究では後者に焦点を当てている。

分散的な制御手法における設計手法には行動規範型設計 [5] と自動設計がある。行動規範型設計ではサブサンプレションアーキテクチャ、確率的有限状態機械、ポテンシャルフィールドなどが制御器に用いられており、その設計は手動で行われている。つまり、群れとしての望ましい振舞いが得られるまで、個ロボットの行動の改良を繰り返す。設

計者は群れの振舞いを直接的に設計できないことから、この設計過程は相当な労力を要する。

自動設計法には集団的振舞いを得る際の設計者の労力を減らしてくれる可能性がある。ここでは、進化ロボティクスアプローチ (ER) とマルチロボット強化学習の 2 つが多く用いられる。ER では人工神経回路網 (ANN) を制御器とし進化計算を用いて設計する機会が多い。しかし、ANN はシミュレーションで得られた結果を実機に実装する際に、その非線形性から所望の振舞いが得られないことが多い。さらに、実機での再調整も難しい。

本研究では、確率的有限状態機械 (PFSM) を制御器として採用する。各状態をモジュールとして実装し、確率的に状態を遷移させる。この状態遷移確率を設計変数とし進化計算を用いて集団的振舞いを設計する。進化型確率的有限状態機械 (EPFSM) アプローチを提案する。得られる PFSM は状態遷移図として表すことができるため、視認性に優れ、人がその意味を理解可能である。また、実機実装が容易であることが期待できる。

空間的組織化行動の中で最も基礎的な要素となるものは集合である。集合行動では一般的に、ある制約条件のもとでロボット群を配置し、空間内でロボット群をお互いに最接近させる。集合行動に分散的な制御手法を適用する場合には、ロボットの制御器は行動規範型設計または ER によって設計されている。多くの研究 [6] がなされているが、群れに関する局所的な情報を取得するセンサの測定範囲によって、1 つのクラスタへの集合が困難になるといった問題が報告されている [7]。

本稿では EPFSM をスワームロボットの集合行動生成に適用し、その性能を検証する。第 2 節では進化計算として適用する粒子群最適化 (PSO) について述べる。第 3 節では PFSM および最適化問題としての定式化について述べる。第 4 節では計算機実験において PSO を用いて PFSM を設計し、集合行動の生成を行い、提案手法の性能を検証する。

2. Particle Swarm Optimization

Particle Swarm Optimization (PSO)[8] は魚群や鳥の群れの社会行動に着想を得た確率的多点探索アルゴリズムであり、その探索性能とアルゴリズムの簡便さからさまざまな分野で適用されている。

2.1 単一目的 PSO

PSO を適用する単一目的最適化問題は以下のように定式化される。

$$\min_{\mathbf{x}} f(\mathbf{x}), \quad (1)$$

ここで、 $\mathbf{x} = (x_1, x_2, \dots, x_L)^T \in R^L$ は L 次元実数値ベクトルである。

PSO では探索空間における解を「粒子」とよぶ。粒子群におけるすべての粒子はそれぞれ位置と速度をもつ。粒子の位置と速度は次式によって更新される。

$$v_{ij}^{k+1} = w \cdot v_{ij}^k + c_1 \cdot \text{rand}() \cdot (pbest_{ij} - x_{ij}^k) + c_2 \cdot \text{rand}() \cdot (gbest_j - x_{ij}^k), \quad (2)$$

$$x_{ij}^{k+1} = x_{ij}^k + v_{ij}^{k+1}, \quad (3)$$

ここで、 $\text{rand}()$ は区間 $[0, 1]$ の一様分布に従う乱数を表し、 w, c_1, c_2 は各項の重みパラメータである。 x_{ij}^k は k 世代目における i 番目の粒子の j 番目の位置成分、 v_{ij}^k は i 番目の粒子の j 番目の速度成分を表す。 $pbest_{ij}$ は i 番目の粒子がそれまでの探索で発見した最良点の j 番目の位置成分であり、 $gbest_j$ は全粒子がそれまでの探索で発見した最良点の j 番目の位置成分である。各粒子は最適化すべき目的関数 $f(\cdot)$ によって評価される。(2) 式の更新式は “inertia weight model” とよばれる最も基礎的なものの 1 つであり、本稿でも採用する。

2.2 多目的 PSO

多目的最適化問題は以下のように定式化される。

$$\min f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_n(\mathbf{x}) \quad (4)$$

多目的最適化問題では目的関数が複数あるため、単一の目的関数では解の優劣を評価できない。そこで一般的に支配関係に基づいて解の優劣を決定する。最小化問題において、

$$f_t(\mathbf{x}_1) \leq f_t(\mathbf{x}_2), \quad \forall t \in \{1, 2, \dots, n\} \quad \text{かつ}, \quad (5)$$

$$f_t(\mathbf{x}_1) < f_t(\mathbf{x}_2), \quad \exists t \in \{1, 2, \dots, n\}$$

が成り立つとき、「 \mathbf{x}_1 が \mathbf{x}_2 を優越する」、もしくは、「 \mathbf{x}_2 が \mathbf{x}_1 に優越される」、という。他のいかなる解にも優越されない解をパレート最適解という。一般に、お互いに優越

されない解は複数個存在し、解は一意には定まらない。全てのパレート最適解が変数空間で形成する集合をパレート最適集合、パレート最適集合の目的関数空間に対する写像をパレートフロントという。語句の定義は [9] に従う。

PSO の更新 ((2) 式) には $pbest_i$ および $gbest$ を設定する必要がある。(3) 式で更新された粒子 x_i が $pbest_i$ を優越する場合に $pbest_i$ を更新する。探索過程で発見された複数のパレート最適解候補はアーカイブとよばれるメモリに保存される。 $gbest$ はアーカイブ内の 1 つの粒子が選択されるのが自然な設定である。この選択はガイド選択とよばれ、いくつかの方法が提案されている [10]。本稿では、アーカイブ内からランダムに選択する方法 [9] を適用する。

アーカイブは以下の方法で更新する。第 k 世代にアーカイブに保存されているパレート最適解候補集合を $S(k)$ とする。また、 $S(k)$ に第 $(k+1)$ 世代において更新された全粒子を加えた集合を $S'(k+1)$ とする。 $S'(k+1)$ 内の粒子の優越関係を評価し、他の粒子に優越されない粒子を選択することでパレート最適解候補集合 $S(k+1)$ を更新し、アーカイブに保存する。なお、アーカイブに保存されているパレート最適解候補集合の数が上限値を超える場合にはその粒子数を削除する必要がある。

3. 確率的有限状態機械

3.1 確率的有限状態機械の定義

SR の研究領域において、確率的状態遷移を伴う有限状態機械であると考えられる制御器を採用している文献が見られる。Brambilla[1] はこの種の制御器を確率的有限状態機械 (probabilistic finite state machines: PFSMs) として分類している。

SR で採用される PFSM では、個ロボットに対し複数の状態を設定する: $S = \{S_1, S_2, \dots, S_m\}$ 。ここで、 S は状態の集合を表し、 m は状態数を表す。この状態は、いわゆる、モジュール [5]、プリミティブ、基本行動 [6] とよばれるものと等しい。状態はある確率で次の状態に遷移する (図 1)。この確率は状態遷移確率行列として表すことができ、以下のように定義される。

$$P = \{p_{ij} \in \mathbb{R} \mid 0 \leq p_{ij} \leq 1, \sum_{j=1}^m p_{ij} = 1, \quad (6)$$

$$i, j \in \{1, 2, \dots, m\}\},$$

ここで、 $P \in \mathbb{R}^{m \times m}$ は行列を表し、要素 p_{ij} は状態 i から状態 j への遷移確率を表す。

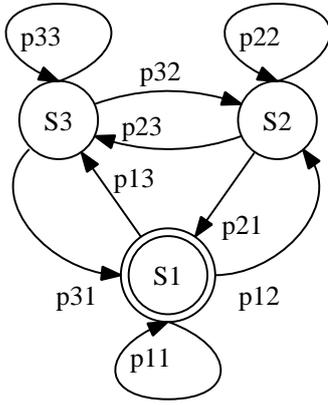


Fig. 1: State transition diagram of the PFMSM

3.2 PFMSM の設計

PFMSM を設計対象とする最適化問題は (6) 式より以下のように定義できる .

$$\min f(p_{ij}), \quad (7)$$

$$\text{subject to } 0 \leq p_{ij} \leq 1, \quad i, j = 1, 2, \dots, m \quad (8)$$

$$\sum_{j=1}^m p_{ij} = 1, \quad i = 1, 2, \dots, m \quad (9)$$

$|P| = m \times m$ であることから, 設計変数の数は $L = m \times m$ となる .

2. で述べた PSO の探索過程において, 初期の粒子群 x^0 の各設計変数は $[0, 1]$ の範囲に一様に分布させる . x^0 および (2)(3) 式によって更新される k 世代における x^k は制約条件 ((8)(9) 式) を満たさない可能性がある . 評価の際に x^k を非実行可能解としないために, これらの制約条件を考慮しなければならない .

本研究ではこれらの制約条件を以下の制約条件に置き換える .

$$0 \leq p_{ij}, \quad (10)$$

$$p_{ij} \leq 1 \cap \sum_{j=1}^m p_{ij} = 1, \quad i, j = 1, 2, \dots, m \quad (11)$$

(10) 式を p_{ij} が満たすために, p_{ij} に以下のような処理を施す .

$$p'_{ij} = \begin{cases} -p_{ij} & \text{if } p_{ij} < 0 \\ p_{ij} & \text{otherwise.} \end{cases} \quad (12)$$

p'_{ij} は次回更新のための設計変数として保存する . つぎに, (11) 式を満たすように, p'_{ij} を以下のように正規化する .

$$p''_{ij} = \frac{p'_{ij}}{\sum_{j=1}^m p'_{ij}}. \quad (13)$$

p''_{ij} は目的関数値の評価にのみ用いて, 設計変数としては保存しない .

4. 制御タスクと目的関数

1. で述べたように, 本稿ではスワームロボットの集合を扱う . 集合問題ではスワームロボットは初期位置から可能な限り互いに近づいて集合することを目的とする . 集合行動はさまざまな応用につながる基礎的要素と考えられており, 集合問題はケーススタディとして多く採用されている [6][7] .

本稿では, 文献 [7] に従って, 集合行動の評価にクラスタ指標 (*cluster metric*) と分散指標 (*dispersion metric*) なる, 2 つの性能指標を採用する .

クラスタ指標は以下のように定式化される .

$$c = \frac{\text{number of robots in the largest cluster at time } t}{N} \quad (14)$$

ここで, 分母の N はロボットの台数を表す . また, 分子の最大クラスタは連結サブグラフの最大値をとるものを差し, 2 台のロボットはその中心間の距離が $4r$ 以下のときに連結しているとみなす . ここで, r はロボットの半径を表す . 本研究では最小化問題を扱うため, 1 つめの目的関数を以下のように与える .

$$f_1 = \frac{1}{c} \quad (15)$$

分散指標は以下のように定式化され, これを 2 つめの目的関数として与える .

$$f_2 = \frac{1}{4r^2} \sum_{h=1}^N \|p_h^{(t)} - \bar{p}^{(t)}\|^2, \quad (16)$$

ここで, $p_h^{(t)}$ は時間ステップ t における h 番目のロボットのタスク環境における位置を表す . $\bar{p}^{(t)}$ は全ロボットの中心位置を表す . この f_2 に関しては各 N に幾何学的な最小値が報告されている [11] . 分散を示す指標であるが, 統計学における「分散」とは定義が若干異なることに注意されたい .

本研究では, t を各試行における最大時間ステップとする . つまり, 各試行の終了時に各目的関数を評価する .

5. 計算機実験

5.1 計算機実験設定

5.1.1 スワームロボットおよび環境の設定

スワームロボットに関する計算機実験の設定は文献 [12] を参考にして以下のように行う . 左右二輪独立駆動型の移動ロボットを動力学シミュレータ ODE を用いてモデル化する (図 2) . 直径 0.17 m, 高さ 0.075 m の台車に, 他ロボットに搭載されている光源の強さを測定する光セン

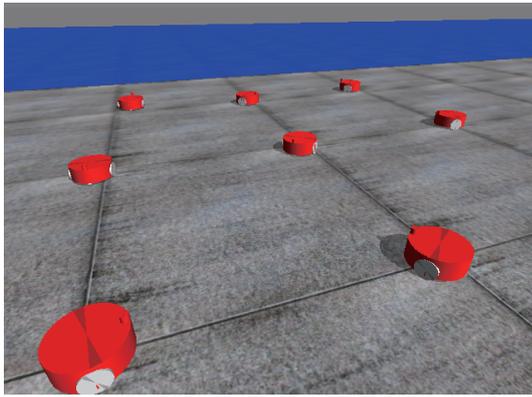


Fig. 2: Setup for swarm mobile robots

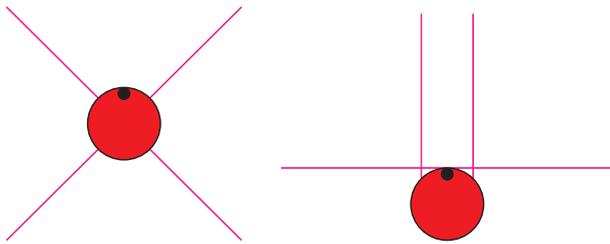


Fig. 3: Distance sensor configurations: (left) sensors for other robots, (right) sensors for obstacles

サ(ロボットの周囲に4個(図3の左), 環境の壁または他ロボットまでの距離を測定する距離センサ(前方に4個(図3の右))を搭載する. 距離センサの計測可能距離は0.3 m, 光センサの計測可能距離は $R \in \{1.0, 1.5, 2.0, \dots, 5.0\}$ [m] とする. 駆動装置としてモータ2個を用いる. スwarmロボットの台数を10とする.

1辺が10 mの壁に囲まれた正方形を実験環境とする(図4). Swarmロボットの初期位置を図4のようにとり, 各ロボットを1m間隔に配置する. 各試行の初期姿勢はランダムに設定する. 各試行は30,000ステップ(300秒に相当)実行する. 実験の試行は10回行う.

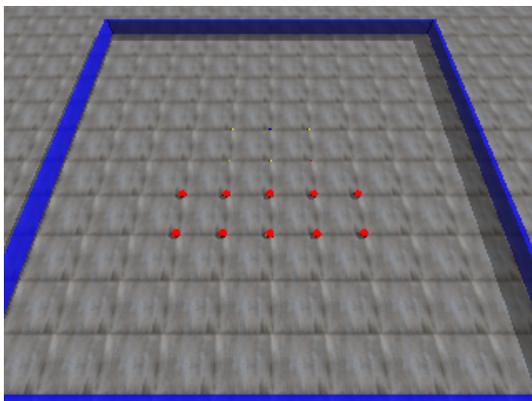


Fig. 4: Set up for computer simulation: Initially, swarm robots are always placed at the center.

5.1.2 制御器

図5に本実験で採用する制御器を示す. 本制御器は文献[6]を参考としている. 制御器はPFISMの各状態における出力(楕円・白抜きで表示), ランダムウォークモジュール, 回避モジュール, 接近モジュール, 停止モジュール(長方形・白抜き)で構成される.

本研究で採用するPFISM(図6)は以下の3つの状態をもつ: $\{Approach, Wait, Repel\}$. $m = 3$ のため, $|P| = 9$ となる. 以下に詳細を述べる.

- *Wait*: ロボットは即時に停止する.
- *Approach*: (i) 光センサが他ロボットの光源を検出した場合, その強度の合計が最も大きい方向に接近する. (ii) 距離センサによって障害物を検出した場合, 回避モジュールを活性化する. (iii) 他ロボットも障害物も検出できない場合, ランダムウォークモジュールを活性化する.
- *Repel*: (i) 他ロボットもしくは障害物を検出した場合, 回避モジュールを活性化する. (ii) 他ロボットも障害物も検出できない場合, ランダムウォークモジュールを活性化する.

ランダムウォークモジュールでは, 移動フェーズと回転フェーズを交互に選択する[12]. 移動フェーズでは, ロボットは左右モータを回転させ前進する(図5における直進モジュール). 移動フェーズの実行時間は6秒に設定した. 回転フェーズでは, $\{45, 90, 135\}$ [°] のうちいずれかの角度と回転方向(時計回り, 反時計回り)をランダムに選択し, あらかじめ測定しておいた, おおよそ, その角度に回転するであろう実行時間でその場で回転する(図5における左右回転モジュール).

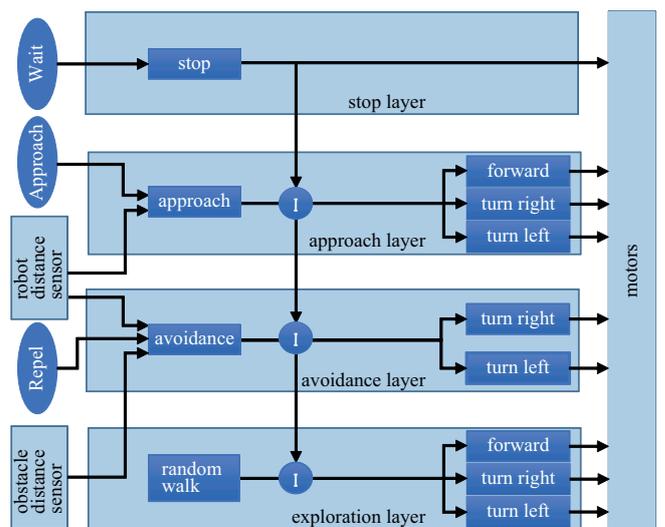


Fig. 5: Subsumption architecture[5] with the PFISM

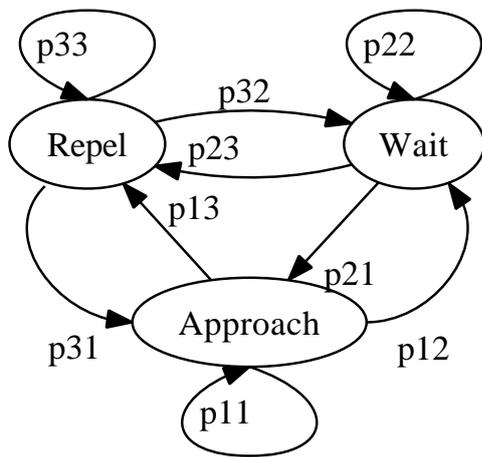


Fig. 6: State transition diagram of the PFISM for aggregation

5.1.3 PSO の設定

3.2 で述べたように、PFISM の状態遷移確率を設計変数とし、PSO を適用する。設計変数の数は $L = 9$ となる。全粒子数を 20 とし、計算機実験を実行する。PSO のパラメータ ((2) 式) は $c_1 = 2.0, c_2 = 2.0, w = 0.5$ とする。全世代数を $k_{\max} = 200$ とする。アーカイブの上限値を 20 とする。各条件に対し、PSO を 10 回実行する。

5.2 計算機実験結果

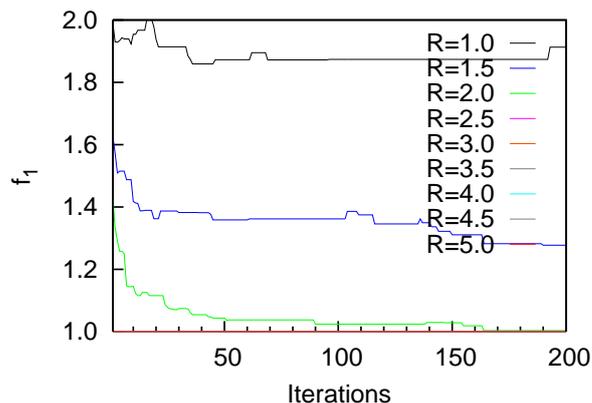
5.2.1 目的関数値

図 7(a), 7(b) に世代数に対する f_1 と f_2 の変化 (10 試行の平均) をそれぞれ示す。光センサの計測可能距離 R の値によって大きく傾向が分かれる。 $R \geq 2.5$ の場合、初期世代から f_1 が 1 を示す¹。これはクラスタ数が 1、つまり、集合行動を達成していることを表している。分散を表す f_2 も早い世代から小さい値をとり、徐々に最適値に近づいている。 $R = 2.0$ の場合にも最終世代付近でクラスタ数は 1 を示す。一方、 $R \leq 1.5$ の場合には、 f_2 は減少するものの、 f_1 はそれほど減少せず、クラスタ数が 1 になることはなかった。

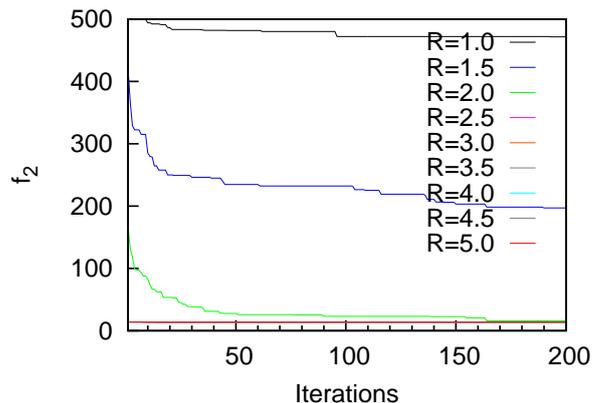
5.2.2 獲得された PFISM の状態遷移図

上述の実験で獲得した制御器の特徴をみる。図 8 に $R = 2.5$ および 5.0 の場合に最も良い性能を示した PFISM の状態遷移図を示す。 $R = 5.0$ の場合には Approach に遷移する確率が高い。ただ、Approach だけに遷移するわけ

¹図 7 では、 $R \geq 2.5$ の場合のグラフは重なっているため、 $R = 5.0$ のみが表示されて見える。



(a) f_1 : $1/\text{MaxClusterRatio}$



(b) f_2 : Dispersion

Fig. 7: Objective function values for each iteration

ではなく、ある割合で Approach から Repel に移行している。これにより、クラスタを形成した後も少し離れたり移動することでより密に集合することが可能になると考えられる。一方、 $R = 2.5$ の場合には Approach に遷移する確率も高いが、 $\text{Repel} \rightarrow \text{Wait}$ の確率が高く、群れへの接近を維持しながら、より動く・離れるという振舞いが備わっていると考えられる。

5.2.3 拡張性に関する検証

上記の $R = 2.5$ および 5.0 の場合に最も良い性能を示した PFISM に関して、ロボット台数を変更した場合の f_2 の値 (10 回平均) を表 1 に示す。theory は文献 [11] で得られている各 N に対する f_2 の幾何学的最小値を示している。なお、 $(R, N) = (2.5, 40)$ を除く各 (R, N) の組合せでクラスタ数は 1 となることを確認している。 $N = 10, 20$ のときには最適値に非常に近い値を示した。また、 $N = 30$ のときにも、 f_2 は最適値よりも大きい値となるが、近い値を示した。 $N = 40$ のときに、 $R = 5.0$ の場合ではクラスタ数は 1 になり、 f_2 も近い値を示している。一方、 $R = 2.5$ の場合ではクラスタ数は 1 にはならず、 f_2 も大きい値となっ

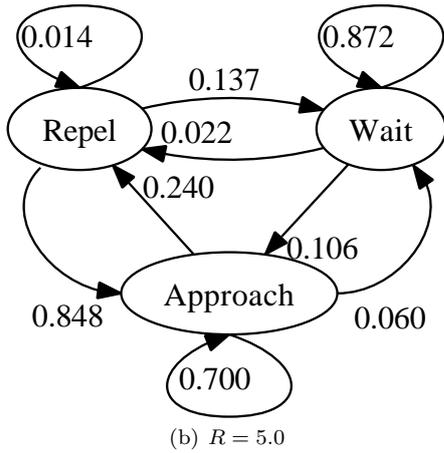
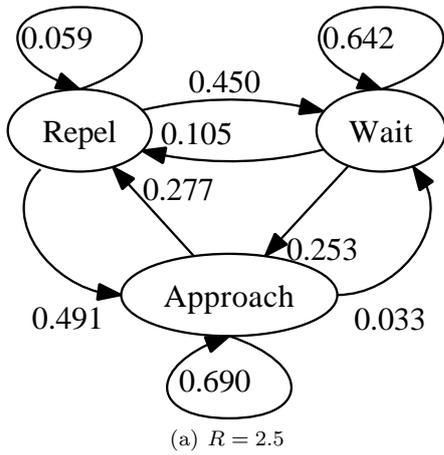


Fig. 8: State transition diagram of the best optimized PFSM

た．図9に $R = 2.5$, $N = 30, 40$ のときのロボットの最終位置を示す．

6. 結言

本稿では，進化型確率的有限状態機械をスワームロボットの集合行動生成に適用し，その性能を検証した．センサの計測可能距離が閾値以上である場合に，群れの数が1となる集合行動の生成が可能であることを確認した．また，設計時に設定した台数以上のロボットでもある程度の台数

Table. 1: Performance of the best controllers on f_2

| | N | | | |
|---------------|-------|-------|--------|---------|
| | 10 | 20 | 30 | 40 |
| <i>theory</i> | 13.50 | 54.65 | 123.5 | 220.33 |
| $R = 2.5$ | 13.93 | 59.99 | 140.80 | 5060.16 |
| $R = 5.0$ | 14.06 | 59.09 | 136.02 | 269.626 |

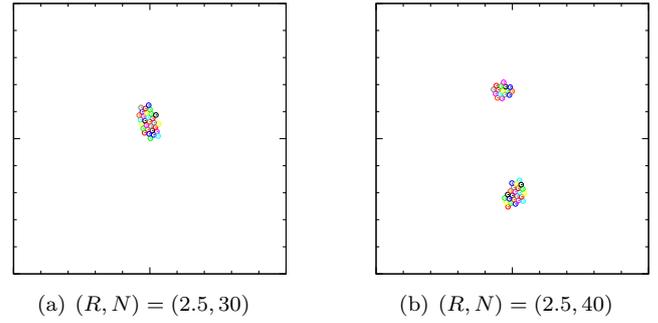


Fig. 9: Final positions of the swarm robots with the best optimized PFSM

までは集合行動が可能であり，拡張性が高いことを確認した．今後は実機実験を通して性能を検証したい．

参考文献

- [1] Brambilla, M., Ferrante, E., Birattari, M., Dorigo, M., “Swarm Robotics: A Review from the Swarm Engineering Perspective,” *Swarm Intelligence*, Vol.7, No.1, pp.1–41, 2013.
- [2] Trianni, V., *Evolutionary Swarm Robotics*, Springer-Verlag 2008.
- [3] Şahin, E., “Swarm Robotics: From Sources of Inspiration to Domains of Application,” *Lecture Notes in Computer Science*, Vol.3342, pp.10–20, 2005.
- [4] Bahçeci, E., Soysal, O., Şahin, E., “A Review: Pattern Formation and Adaptation in Multi-Robot Systems”, Technical Report CMU-RI-TR-03-43, CarnegieMellon University, 2003.
- [5] Brooks, R., “A Robust Layered Control System for a Mobile Robot”, *IEEE Journal of Robotics and Automation*, Vol.2, No.1, pp.14–23, 1986.
- [6] Soysal, O., Bahçeci, E., Şahin, E., “Aggregation in Swarm Robotic Systems: Evolution and Probabilistic Control”, *Turkish Journal of Electrical Engineering and Computer Sciences*, Vol.15, No.2, pp.199–225, 2007.
- [7] Gauci, M., Chen, J., Li, W., Dodd, T. J., Groß, R., “Self-Organised Aggregation without Computation”, *The International Journal of Robotics Research*, Vol. 33, No. 9, pp. 1145–1161, 2014.
- [8] Kennedy, J., Eberhart, R.C., “Particle Swarm Optimization”, *In Proceedings of IEEE International Conference on Neural Networks*, pp. 1942–1948, 1995.
- [9] 増田和明, 栗原謙三, “多目的 Particle Swarm Optimization に基づく制約条件付き大域的最適化手法”, *電気学会論文誌 C*, Vol. 131, No. 5, pp. 990–999, 2011.
- [10] 内種岳詞, 畠中利治, “単一の目的関数に基づくガイド選択による多目的 PSO”, *進化計算学会論文誌*, Vol. 3, No. 3, pp.155–162, 2012.
- [11] Graham, R.L., Sloane, N.J.A., “Penny-Packing and Two-Dimensional Codes”, *Discrete Computational Geometry* 5: p.1–11, 1990.
- [12] Katada, Y., Nishiguchi, A., Moriwaki, K., Wataabe, R., “Swarm Robotic Network Using Levy Flight in Target Detection Problem”, *Artificial Life and Robotics*, Volume 21, Issue 3, pp. 295–301, 2016.