The 33rd Annual Conference of the Japanese Society for Artificial Intelligence, 2019

Deep Neuroevolution によるロボティックスワームの二点間往復タスクにおける群れ行動の生成

Generating Collective Behavior of a Robotic Swarm in a Two-landmark Navigation Task with Deep Neuroevolution

森本 大智 *1 Daichi MORIMOTO　平賀 元彰 *1 Motoaki HIRAGA　大倉 和博 *1 Kazuhiro OHKURA　松村 嘉之 *2 Yoshiyuki MATSUMURA

*1広島大学 Hiroshima University　*2信州大学 Shinsyu University

Deep reinforcement learning has provided outstanding results in various applications. Deep neural networks are usually trained by gradient-based methods. However, when deep reinforcement learning is applied to a robotic swarm, that is composed of many robots, it is difficult to design reward functions that lead to a desired collective behavior. In this paper, we applied deep neuroevolution, which is a technique to optimize deep neural networks with artificial evolution, to design controllers of a robotic swarm. Deep neuroevolution is expected to evolve deep neural networks to different reward/fitness landscapes because it optimizes with population-based and gradient-free methods. This paper shows that the controllers designed with deep neuroevolution give robustness to different reward settings compared to deep reinforcement learning.

1. はじめに

Swarm Robotics(SR) は多数のロボットからなる集団に，ロボット間あるいはロボットと環境間の局所的な相互作用によって所望の群れ行動の生成を目指す研究分野である [Sahin 2004]. SR では群れを構成するロボットの台数が増加するに従い，設計者が各ロボットの行動を定義して制御器を設計することが困難となる．そのため，SR では強化学習や進化ロボティクスを制御器設計に適用する試みがなされている [Brambilla 13].

強化学習の分野では近年，深層ニューラルネットワーク (Deep Neural Network,DNN) を組み合わせた深層強化学習 (Deep Reinforcement Learning,DRL) が様々な問題に適用され，良好な成績を記録している [Mnih 15, Lillicrap 15]．一般的に DRL における DNN の学習には，重みの勾配を用いた単点探索型の勾配降下法が用いられる．そのため，多峰性の目的関数における局所解や異なる報酬設定に対応するのが難しい．したがって，DRL をロボティックスワームに適用する場合，性能を向上させるために適切な報酬設定が必要となる．

本稿では Deep Neuroevolution(DNE) によるロボティックスワームの制御器設計を行う．DNE は DNN を進化計算で学習させる手法である．DNE は個体群ベースの探索によって DNN を学習させるため，報酬設定の変化に対して頑健であることが期待される．本稿では異なる報酬条件下で，DRL および DNE によってロボティックスワームの制御器である DNN を設計し，DRL と DNE の報酬設定に対する頑健性，およびロボティックスワームの群れ行動の比較を行う．

2. Deep Neuroevolution

Deep Neuroevolution(DNE) は進化計算によって DNN を学習させる手法である．一般的に DRL における DNN の学習は，重みの勾配を用いた単点探索型の勾配降下法で行われる．そのため多峰性の目的関数における局所解や異なる報酬設定に対応するのが難しい．対して DNE では個体群ベースの探索のためこれらの問題に関して頑健な DNN の学習が期待される．

連絡先: 森本大智，広島大学，morimoto@ohk.hiroshima-u.ac.jp

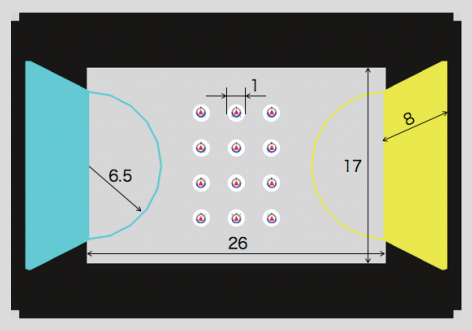


図 1: 実験環境

DNE の先行研究として [Salimans 17, Such 17] がある．Saliman らは三層の畳み込み層と二層の全結合層からなる DNN を Natural Evolution Strategy を用いて学習させた [Salimans 17]．また Such らは同様の構造を持つ DNN を交叉を用いず突然変異のみを用いる Genetic Algorithm によって学習させた [Such 17]．これらの先行研究は Atari 2600 などの DRL の問題に適用されており，本稿のロボティックスワームのような多数のエージェントを扱う問題には適用されていない．

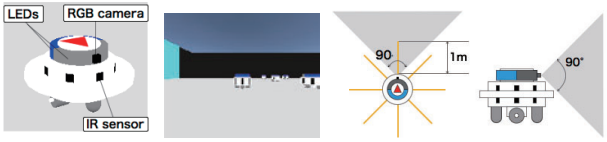
3. 計算機実験

3.1 二点間往復タスク

実験には二点間往復タスクを用いる．このタスクの目的は二つのターゲットエリア間の往復である．実験環境を図 1 に示す．環境の両端に黄色とシアンのランドマークを配置する．ランドマークの中心から半径 6.5m の範囲がターゲットエリアである．ロボットは目標とするターゲットエリアに到達時，目標をもう一方のターゲットエリアに変更する．ロボットは各エピソード開始時に環境の中央に格子状に配置される．

3.2 ロボット

図 2(a) に実験に用いた移動ロボットを示す．ロボットの直径は 1m であり左右の二つの車輪とモータによって駆動する．



(a) 移動ロボット (b) カメラ画像 (c) センサ設定

図 2: ロボットの設定

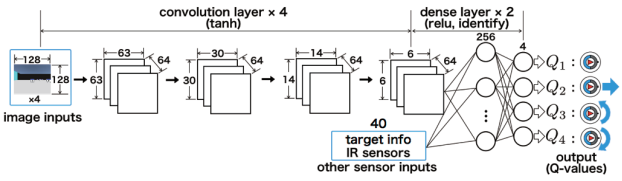


図 3: ロボットの制御器

最高移動速度は 1 m/s である．本実験ではロボットが取る行動を停止，直進，左回転，右回転の 4 種類とする．ロボットはセンサとしてカメラ 1 個と距離センサ 8 個を搭載している．カメラは 128×128 pixels の RGB 画像を生成する．図 2(b) にカメラから得られる画像の例を示す．距離センサは測定範囲内の物体までの距離を測定する．各センサの測定範囲を図 2(c) に示す．ロボットの上部に搭載された LED は後方のみ点灯し，黄色のランドマークを目指す場合は赤色，シアンの場合は青色を点灯する．

3.3 制御器

ロボットの制御器として図 3 に示す DNN を用いる．制御器は入力層，4 層の畳み込み層，全結合層，出力層からなる．制御器への入力はカメラ画像，距離センサ情報，目標ランドマーク情報であり，それぞれ過去 4 タイムステップまでの情報が入力される．畳み込みに使用するフィルタはサイズを 4 × 4，チャネル数をそれぞれの層の入力に対応させたものを 64 種類用意する．ストライドは 2 とし，これらフィルタに関する設定は 4 層の畳み込み層で共通とする．また各畳み込み層の直前に Batch Normalization 層を加える．

3.4 実験設定

ここでは DRL と DNE における共通の実験設定について述べる．実験の最終世代数は 500 とし，1 世代は 1000 タイムステップとする．制御器は 24 個とし，DRL ではこれらを独立に学習させ，DNE では進化アルゴリズムによって学習させる．またロボットに対する報酬は以下のものを用いる．

$$\begin{aligned} r_{d,i,t} &= 5 \times (d_{i,t-1} - d_{i,t}) \\ r_{e,i,t} &= 5 \\ r_{c,i,t} &= -5 \end{aligned} \tag{1}$$

ここで i はロボットの番号，t はタイムステップを表す．$r_{d,i,t}$ はロボットが目標ランドマークに近づいた距離に応じて与えられる．$d_{i,t}$ はタイムステップ t におけるロボット i の重心とランドマークの重心間の距離を示す．$r_{e,i,t}$ はターゲットエリアへの到達時に与えられる．$r_{d,i,t}$ と $r_{e,i,t}$ は目標ランドマークがカメラに映っていた場合のみ適用される．また $r_{c,i,t}$ は距離センサが他のロボットや壁との接触を検知した際に与えられる．これらの報酬を組み合わせた二つの条件下で実験を行う．一つ目の条件では $r_{d,i,t}$，$r_{e,i,t}$，$r_{c,i,t}$ の総和によって報酬を決定する．1 タイムステップにおける報酬 R_t は以下の式で与えられ，これを「報酬設定 (i)」とする．

$$R_t = \sum_i (r_{d,i,t} + r_{e,i,t} + r_{c,i,t}) \tag{2}$$

二つ目の条件では $r_{e,i,t}$ のみによって報酬を決定する．1 タイムステップにおける報酬 R_t は以下の式で与えられ，これを「報酬設定 (ii)」とする．

$$R_t = \sum_i r_{e,i,t} \tag{3}$$

またロボットの各タイムステップにおける行動は ϵ-greedy 法によって決定される．第 1 エピソードでは $\epsilon = 1$ に設定しランダムに行動を選択する．また 1 エピソード目は制御器の学習を行わない．2 エピソード以降は $\epsilon = 0.1$ に固定し行動を選択しつつ，制御器の学習を行う．

3.5 制御器の学習

DRL

DRL のアルゴリズムとして DQN[Mnih 15] を用いる．DQN では Experience replay によって蓄積した環境の遷移情報を再生し毎タイムステップ制御器の重みを更新する．保有する経験はロボット台数 ×1 エピソードのタイムステップサイズ分であり，全ロボットの経験が共有される．学習時のミニバッチサイズは 32 とし，optimizer には RMSpropGraves[Graves 13] を用いる．

DNE

DNE の場合は 1 エピソード間に獲得した報酬の和 $\sum_t R_t$ を制御器の適応度とし，エピソード終了時に制御器の更新を行う．制御器の進化アルゴリズムとして文献 [Such 17] を参考としたものを用いる．各エピソード終了時にトーナメント選択により親個体を選択する．本実験ではトーナメントサイズを 2 とする．また，エリート選択により最高の適応度を記録した制御器を一つ，遺伝的操作を行わずに次世代に引き継ぐ．選択した親個体に対し突然変異を適用し次世代個体を生成する．突然変異操作は以下の式で表される．

$$\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \boldsymbol{\epsilon}\sigma \tag{4}$$

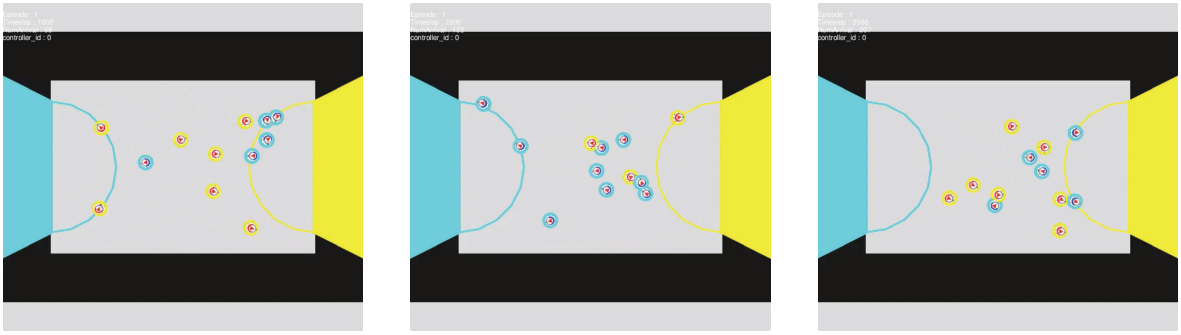
ここで $\boldsymbol{\theta}$ は DNN のパラメータベクトル，$\boldsymbol{\epsilon}$ は $\boldsymbol{\theta}$ と同じ要素数を持つノイズベクトル，σ は遺伝子に与える揺動のスケーリングパラメータである．$\boldsymbol{\epsilon}$ の各要素は標準正規分布よりサンプリングされる．また本実験では σ=0.02 とする．進化させる DNN のパラメータは全結合荷重値と Batch Normalization 層のアフィン変換パラメータとする．

4. 結果

4.1 報酬設定 (i)

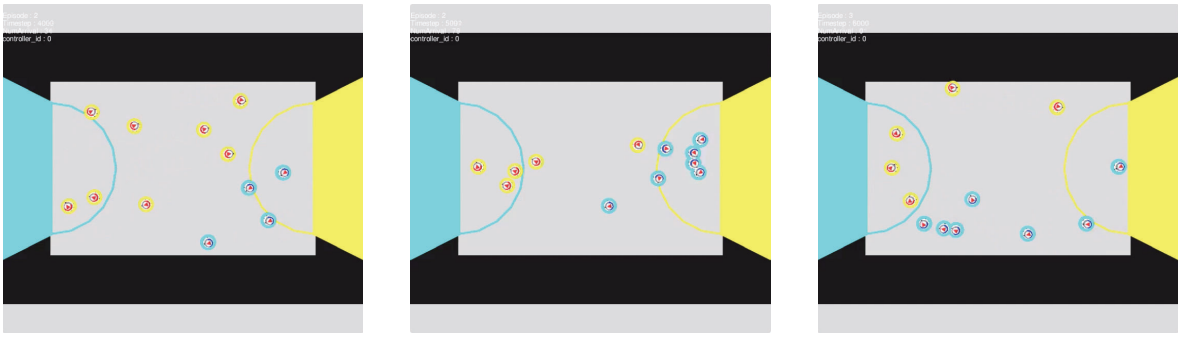
学習時におけるロボットの到達回数の推移を図 4 に示す．グラフは実験 3 試行分のデータを示し，平均値，最大値，標準偏差は 72 個の制御器のデータから算出されている．DRL ではおよそ 100 世代にかけて到達回数が急激に上昇しその後ほぼ一定の値を示す．DNE では 60 世代付近まで平均値がほぼ上昇せず，その後 DRL に比べ比較的緩やかに上昇する．

次に学習中に記録された制御器を用いてロボティックスワームの群れ行動生成を行なった結果を図 6，図 7 に示す．DRL の場合，ロボットは他のロボット間を抜け，それぞれのロボットが独立にターゲットエリアを目指す振る舞いを示す．対して DNE の場合，ロボットは他のロボットを追従し環境内を輪を描くように移動する振る舞いを示す．



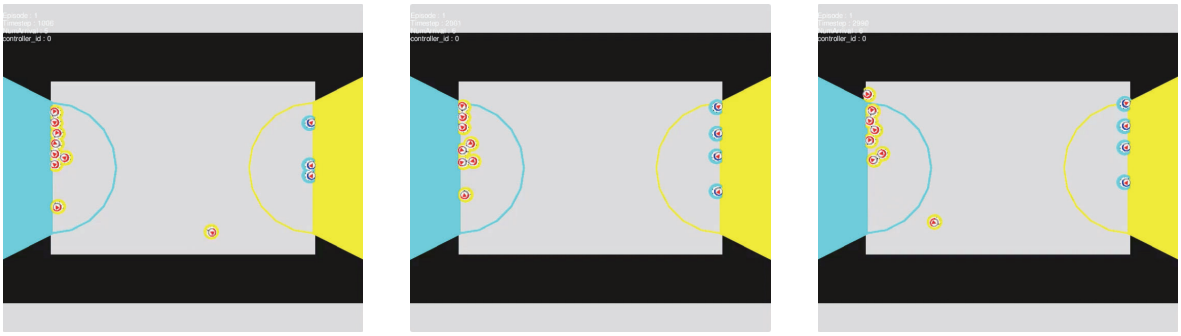
(a) 1000 time steps (b) 2000 time steps (c) 3000 time steps

図 6: 報酬設定 (i) の条件下で DRL が生成した群れ行動



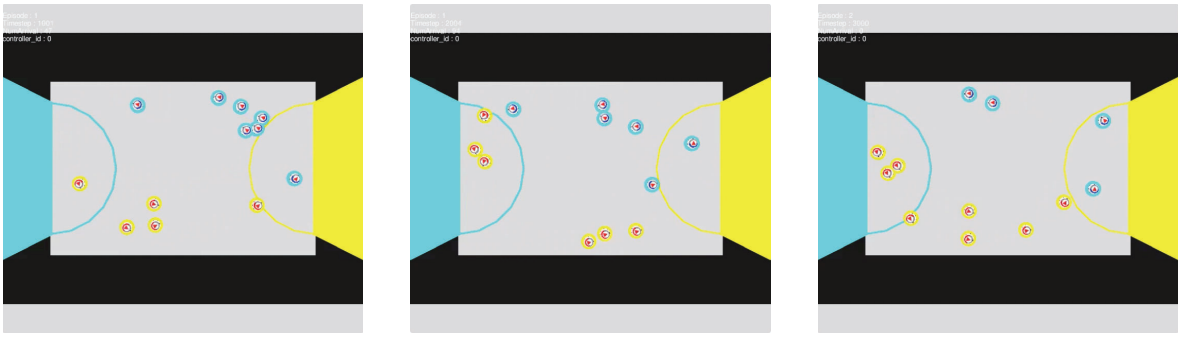
(a) 1000 time steps (b) 2000 time steps (c) 3000 time steps

図 7: 報酬設定 (i) の条件下で DNE が生成した群れ行動



(a) 1000 time steps (b) 2000 time steps (c) 3000 time steps

図 8: 報酬設定 (ii) の条件下で DRL が生成した群れ行動



(a) 1000 time steps (b) 2000 time steps (c) 3000 time steps

図 9: 報酬設定 (ii) の条件下で DNE が生成した群れ行動

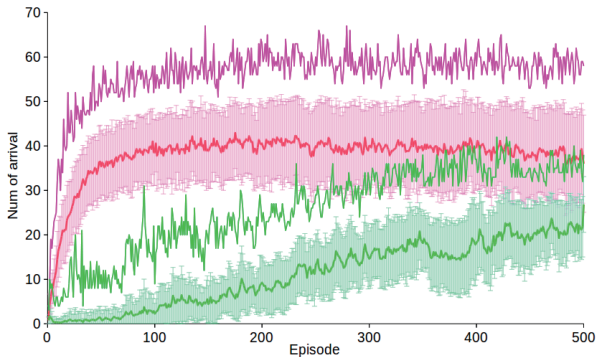


図 4: 報酬設定 (i) における到達回数の推移

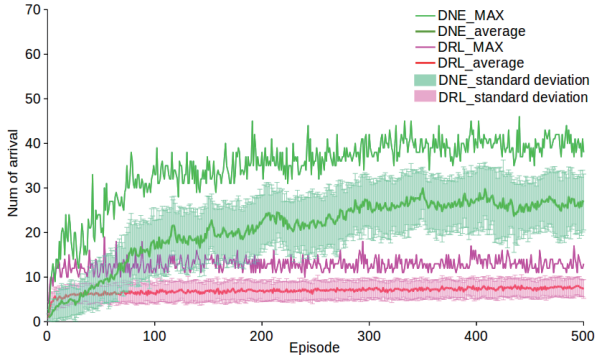


図 5: 報酬設定 (ii) における到達回数の推移

4.2 報酬設定 (ii)

報酬設定 (i) の場合と同様に到達回数の推移を表したグラフを図 5 に示す．DRL では報酬設定 (i) と同様に到達回数の急激な上昇が見られる．しかし，定常時の平均値，最大値は報酬設定 (i) の場合と比べ低下している．DNE では，報酬設定 (i) で見られた 70 世代付近までの停滞が見られず，平均値は学習初期からほぼ単調に増加している．

次に報酬設定 (i) の場合と同様にロボティックスワームの群れ行動生成を行なった結果を図 8，図 9 に示す．DRL の場合，ロボットはターゲットエリアに侵入後ランドマークに衝突し，その後ほぼ移動を行わない．対して DNE の場合，報酬設定 (i) と同様に環境内を輪を描くように移動する振る舞いを示す．

5. 考察

報酬設定 (i) において DNE は，学習初期の 60 世代付近まで到達回数の上昇が滞っている．これは報酬設定 (i) における罰則 $r_{c,i,d}$ によるものであると考えられる．ターゲットエリアへの到達報酬 $r_{e,i,t}$ が発生する頻度は $r_{d,i,t,}$，$r_{c,i,t}$ よりも低い．対して $r_{d,i,t,}$，$r_{c,i,t}$ は条件を満たせば毎タイムステップ発生する可能性がある．またロボットの最高速度を考慮すると，1 タイムステップに発生する $r_{d,i,t,}$ の最大値は 1.0 となり，罰則 $r_{c,i,t}$ の五分の一である．このことから，学習初期において発生する報酬は $r_{c,i,t}$ が支配的であると考えられ，進化の過程において到達回数が多い制御器よりも，衝突を回避する制御器が有利となることで到達回数の停滞が起こると考えられる．

また，報酬設定 (ii) における DRL の到達回数は報酬設定 (i) と比較して低下している．これは与えられる報酬がスパースになるためと考えられる．上述の通り，ターゲットエリアへの到達報酬 $r_{e,i,d}$ は発生する頻度が低い．そのため Experience Replay に用いる記憶領域において，報酬に関する情報が乏しくなることで性能が低下すると考えられる．よって，より大きい記憶領域を用い，Prioritized Experience Replay 等のアルゴリズムを用いることで性能の向上が期待される．

また，本実験で用いた二つの報酬条件において，DNE は最大値やロボットの振る舞いにおいて近しい性能を示したことから，報酬条件に対してより頑健な学習が行えると考えられる．

6. おわりに

本稿では Deep Neuroevolution をロボティックスワームの制御器設計に適用し，二点間往復タスクにおいて群れ行動の生成を行なった．Deep Neuroevolution を適用したロボティックスワームは二つの異なる報酬条件下においてタスクの達成が可能であることを示した．

今後の展望として，より複雑な報酬条件下でロボティックスワームの異なる振る舞いが得られるか実験を行う．また，協調搬送タスクや経路形成タスクに Deep Neuroevolution を適用した場合にタスク達成が可能であるか実験を行う．

参考文献

[Sahin 2004] Erol Sahin: Swarm robotics: from sources of inspiration to domains of application, *Swarm Robotics: SAB 2004 International Workshop*, Vol. 3342 of *Lecture Notes in Computer Science*, pages 10-20. Springer, 2005.

[Brambilla 13] Manuele Brambilla, Eliseo Ferrante, Mauro Birattari, and Marco Dorigo :Swarm robotics: a review from the swarm engineering perspective, *Swarm Intelligence*, Vol. 7, pages 1-41, 2013.

[Mnih 15] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al.:Human-level control through deep reinforcement learning, *Nature, 518(7540):529*, 2015.

[Lillicrap 15] Timothy P Lillicrap, Jonathan J Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, Daan Wierstra :Continuous control with deep reinforcement learning, *arXiv preprint arXiv:1509.02971*, 2015.

[Salimans 17] Tim Salimans, Jonathan Ho, Xi Chen, Szymon Sidor, Ilya Sutskever :Evolution strategies as a scalable alternative to reinforcement learning, *arXiv preprint arXiv:1703.03864*, 2017.

[Such 17] Felipe Petroski Such, Vashisht Madhavan, Edoardo Conti, Joel Lehman, Kenneth O Stanley, Jeff Clune :Deep neuroevolution: genetic algorithms are a competitive alternative for training deep neural networks for reinforcement learning, *arXiv preprint arXiv:1712.06567*, 2017.

[Graves 13] Alex Graves:Generating sequences with recurrent neural networks, *arXiv preprint arXiv:1308.0850*, 2013.