

複数の問題を取り扱うロボットの行動学習の基礎的検討

A Fundamental Study on Learning Robot Behaviors dealing with Multiple Tasks

曾我 紗知子

Sachiko Soga

小林 一郎

Ichiro Kobayashi

お茶の水女子大学大学院人間文化創成科学研究科理学専攻

Advanced Sciences, Graduate School of Humanities and Sciences, Ochanomizu University

The tasks in the real world usually contain multiple problems, therefore, even a robot has to reasonably solve multi-objective problems in order to achieve complex tasks. In the field of evolutionary computation, research on multi-objective optimization has recently been progressive. In this study, we incorporate multi-objective optimization problem into the learning of a robot controller, and consider the learning of a robot's behaviors for multi-objective problems.

1. はじめに

実世界における作業は複数の問題を含んでいることが多く、ロボットの行動において複雑な課題を遂行するためには、多目的の問題を合理的に解決する必要がある。近年、そのような問題に対処するために進化計算において多目的最適化問題が盛んに研究されている。このような背景から本研究では、多目的最適化手法をロボットコントローラの行動学習に取り入れて、多目的問題とされる課題に対するロボットの行動学習について考察を行う。

2. 関連研究

ロボットのコントローラを獲得する手法は多数考えられている。その中でも、コントローラをニューラルネットワークによって構成し、学習することは有効な手段であると示されている [1]。ニューラルネットワークでの学習は、一般的にバックプロパゲーション法を利用する [2][3][4]。一方、ロボットコントローラをニューラルネットワークで構成し、ネットワークの結合荷重を遺伝的アルゴリズムを用いて進化的に獲得する研究も多くなされている [5][6][7][8]。Montana ら [9] はバックプロパゲーション法ではなく、遺伝的アルゴリズムで行うことを提案しており、また、高橋らの研究 [10] では、ある特定の問題を学習させる場合、従来ではニューラルネットワークの設計者が経験や知識等で構造やパラメータを決定していたが、遺伝的アルゴリズムを用いることで、あまり経験を必要とせず、収束速度と収束の安定性が共に満足できることを示している。また、片田 [11] は、遺伝的アルゴリズムを多目的遺伝的アルゴリズムに変更して、ロボットコントローラの学習を行い、行動知識の獲得を示している。東原ら [12] は、多目的遺伝的アルゴリズムを拡張させた GA を提案し、経路追従タスクとゴミ拾いタスクを同時に達成させる行動知識の獲得を行っている。これらの研究では、複数の目的を達成するような行動知識を獲得することができているが、動的な障害物によって絶えず変化し続けている環境での実験は行われていない。本論文では、最初に単目的の課題を多目的遺伝的アルゴリズムを適用することにより解決できることを示し、その後、動的な対応を必要と

する課題を含む多目的の課題を多目的遺伝的アルゴリズムで解決できることを示す。

3. 学習環境

本研究では、Web 上でオープンソースとなっている、進化ロボティクスの研究・教育用 3D シミュレータ Simbad[13] を利用して、ロボットコントローラの学習を行う。Simbad は、ロボットの他、壁や箱などの障害物を設置することができ、様々な環境の下でロボットコントローラの学習を行うことができる。ロボットには、視覚センサ (カラー単眼カメラ)、ソナー、光センサ、衝突検出用のバンパーが取り付け可能となっており、制御対象としては、左右の車輪の速度、回転速度、回転角度である。今回のシミュレーションで利用するのは、センサ (ソナー) と速度、回転速度である。

3.1 コントローラの構成

ロボットのコントローラは、図 1 に示すニューラルネットワークで構成され、遺伝的アルゴリズムによって進化させる。

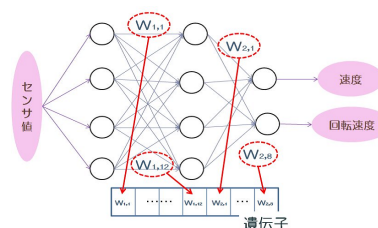


図 1: ロボットコントローラ

コントローラは、単一方向にのみ信号が送られるフィードフォワード型の入力層 4、中間層 4、出力層 2 のニューラルネットワークで構成する。ロボットが取得するセンサ値を入力とし、速度と回転速度を出力とする。遺伝的アルゴリズムを利用する際に必要となる遺伝子には、入力層、中間層、出力層全てのノード間の 24 個の結合荷重の値を遺伝子座に入れる。

4. 単目的行動学習

初めに、後述する多目的問題の課題を達成させるために必要となるロボット同士の衝突を回避する、単目的の行動学習を行い、衝突回避を行うコントローラを獲得する。

連絡先: 曾我紗知子, お茶の水女子大学大学院人間文化創成科学研究科理学専攻情報科学コース小林研究室, 〒112-8610 東京都文京区大塚 2-1-1, Tel.03-5978-5708, soga.sachiko@is.ocha.ac.jp

4.1 学習仕様

学習を行う環境を図 2 に示す．この環境は壁と壁との距離をロボット 2 台並べた直径の約 1.7 倍とし，ロボット 2 台がすれ違うのに十分な長さに設定している．ロボット同士の衝突も検知するように設定し，衝突した際は適応度を 0 にする．

実験環境に存在する 4 台のロボットの内，3 台のロボットはニューラルネットワークの結合荷重を $[-1.0, 1.0]$ のランダムな値に設定したコントローラで動作させ，残りの 1 台のロボットコントローラを壁にもロボットにも衝突しない様に学習させる．学習をしない 3 台のロボットは，壁に衝突してもランダムに方向転換をし，学習しているロボットがリセットされるまで動き続けるように設定した．

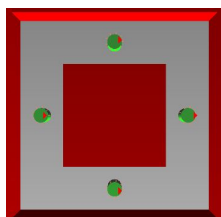


図 2: 複数台ロボットの環境

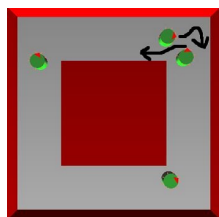


図 3: 衝突回避

学習には遺伝的アルゴリズムを利用する．結合荷重は $[-1.0, 1.0]$ の実数値であり，実際にロボットを動作させ，設定した適応度関数に従って適応度を計算する．次にルーレット選択で次世代の個体を選ぶが，更にエリート保存選択を行っている．その後，交叉率 0.6 で，一点交叉を行う．最後に突然変異率 0.1 で，遺伝子座の値を $[-1.0, 1.0]$ のランダムな値に変更する．これらを設定した世代数分繰り返す．

4.2 適応度関数

学習目的は，障害物に衝突しないこととし，文献 [14] における，壁への衝突回避問題で定義された適応度関数を参考にし，本実験における適応度関数を以下の様に設定した．

$$f = \sum \{v + (\pi - |r|) + s + k\} + 10 \times d \quad (1)$$

- f : 適応度
- v : 速度
- $|r|$: 回転速度の絶対値
- s : センサ値の平均
- k : センシング 1 ステップ間の移動距離
- d : 総移動距離

適応度関数は，ロボットが大きく回転せず，高速に前進し，次のセンサ情報が入力されるまでの移動距離と総移動距離が長くなる場合に適応度が高い値をとるように設定される．コントローラは，センシングの 1 ステップ毎に入力値を代入し，出力値を計算している．それによって，適応度関数も 1 ステップ毎に累積している．設定したステップ数が終了するか，ロボットの速度が 0，または 1 ステップ間の移動距離が 0 になった時点で，適応度関数の累積計算を終了し，最後に総移動距離を足し合わせている．長い距離を移動した場合に適応度が高くなるよう，経験的に移動距離に 10 の値を掛けている．また，壁に衝突した際には適応度を 0 にし，終了条件が満たされるまで計算を続ける．終了条件の 1 つであるステップ数は 2500 と設定した．

4.3 学習結果

100 世代までの学習では，学習した環境 (図 2 参照) のスタート地点から壁に衝突しない半径で回転をする学習をしていた．

200 世代まで学習させ，その学習したコントローラ 1 つを 4 台全てのロボットに組み込んで動作確認をしたところ，ロボットは壁とロボット同士の両方に衝突しないで動作した．ロボット同士が衝突しそうになると，速度を緩め，お互いに横に避けて進む動き (図 3 参照) を見せた．

200 世代以上の学習を行うと，100 世代までの学習と同じ様に，ロボットは回転を続ける動作を引き起こしてしまった．また，100 世代の時より速度が速くなっており，2500 ステップの中で適応度をより高くしようと，総移動距離を長くするために，その場で回転を続けることと，速度を速くすることを学習したのだと考えられる．

5. 多目的の基本行動学習

ロボットの行動学習において，より複雑な行動を学習するためには，複数の目的から成るタスクを扱う必要がある．そこで，適応度関数を複数にし，多目的遺伝的アルゴリズムを利用する．まず初めに，我々が持つロボットの行動学習環境において，片田 [11] の実験を再現する実験を行い，ロボットの挙動を確認した．

学習させる環境を図 4 に示す．ここで，ロボットは与えられた環境において，壁に衝突せずに巡回する行動を学習することを目的とする．本来，この課題は単目的の行動としても学習可能であるが，単目的の課題を敢えて複数の目的から構成される課題と捉え，多目的の基本行動学習の検証を行う．この際，「壁に衝突せずに巡回する」という行為を「直進する」および「壁からの距離を保つ」という，2 つの目的を同時に満たす多目的問題を解決するものとして行動学習を行う．

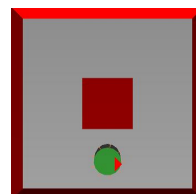


図 4: 多目的の学習環境

5.1 学習アルゴリズム

学習には，多目的最適化問題を解決する手法で，広く利用されているものの 1 つである多目的遺伝的アルゴリズムの NSGA-II [16] を用いる．図 5 に多目的最適化におけるロボットコントローラの学習概要を示す．

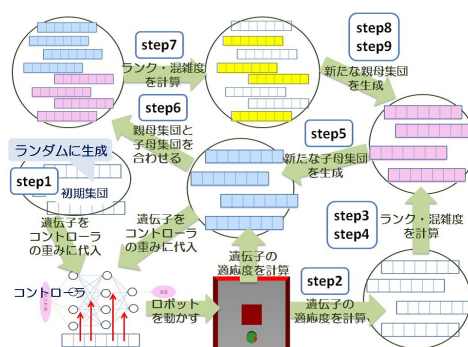


図 5: 多目的最適化におけるロボットコントローラの学習

コントローラの学習アルゴリズムを図 5 に従って，以下に示す．

step 1. 初期集団の生成

結合荷重の数 24 個を $[-1.0, 1.0]$ の実数値でランダムに

発生させ、1つの遺伝子の中に代入する。この個体を50個生成し、 P_t とする(この時、 $t=0$)。

step 2. 適応度の計算

コントローラのノード間の結合荷重に1つの個体の値を代入し、シミュレータ上でロボットを動作させ、設定した2つの適応度関数に従って適応度をそれぞれ計算する。これを50個全ての個体で行う。

step 3. ランク付け

個体群をランク毎に分類。
以下にランク付けのアルゴリズムを示す。

- step i. 各個体に対して、支配されている個体の数を数える。
- step ii. 支配されている個体が0である個体をランク r とする(初期値は $r=1$ とする)。
- step iii. step ii. でランク付けされた個体を除く。
- step iv. $r=r+1$ として step i. へ戻る。step i. ~ step iv. を全ての個体がランク付けされるまで繰り返す。

step 4. 混雑度計算

個体群に混雑度をそれぞれ与える。
以下に混雑度の計算のアルゴリズムを示す。

- step i. ランクが r である個体を適応度の値が悪い順にソートする(初期値は $r=1$ とする)。
- step ii. 適応度が最大と最小のそれぞれの個体に混雑度として無限の値を与える。
- step iii. step ii. で値が与えられた個体を除いた残りの個体に対して以下の式で混雑度を与える。

$$d_i = \sum_{m=1}^M \frac{f_m^{i+1} - f_m^{i-1}}{f_m^{max} - f_m^{min}}$$

d_i : ランク r の中でソートした個体の i 番目の個体
 m : 適応度の番号 ($m=1, 2, \dots, M$)
 f_m^i : i 番目の個体の m 番目の適応度の値

- step iv. $r=r+1$ とし、step i. へ戻る。全ての個体に混雑度が与えられるまで step i. ~ step iv. を繰り返す。

step 5. 新たな子母集団 Q_t を生成

親母集団 P_t を基に、混雑度トーナメント選択、交叉率0.6で交叉、突然変異率0.1で突然変異を行い、個体数50の新たな子母集団 Q_t を生成し、 Q_t に対して step 2. を実行する。

step 6. $R_t = P_t \cup Q_t$ を生成

親母集団 P_t と子母集団 Q_t を合わせて、個体数100の新たな母集団 R_t を生成する。

step 7. R_t に対して step 3. と step 4. を実行

R_t の100個の個体に対してランク付けと混雑度計算を行う。

step 8. 新たな親母集団 P_{t+1} を生成

r をランクとし、その初期値を $r=1$ とする。 R_t の中からランクが小さいものから順に P_{t+1} の個体数が50を超えない条件の下で、新しい母集団 P_{t+1} に加える。 $r=r+1$ とし、step 8. を繰り返す。 P_{t+1} の個体数が50より大きくなる場合は、 P_{t+1} に加えずに step 9. へ移動する。

step 9. P_{t+1} の個体数を50にする

R_t において、 P_{t+1} の個体数が50を超える最小のランク r を持つ個体のうち、多様に広がっているものを $50 - |P_{t+1}|$ 個 P_{t+1} に加え、 P_{t+1} の個体数を50にする。

step 10. 世代の更新または終了

step 5. ~ step 9. を設定された世代数になるまで繰り返す。設定された世代数になったら終了する。設定された世代数に満たなかったら $t=t+1$ とし step 5. へ戻る。

5.2 適応度関数

目的は、直進移動と衝突回避の2目的とする。4.2節の適応度関数を利用し、以下の様に設定する。

$$f_1 = \sum \{v + (\pi - |r|)\} \quad (2)$$

$$f_2 = \sum s \quad (3)$$

f_1, f_2 : 適応度 v : 速度
 $|r|$: 回転速度の絶対値 s : センサ値の平均

式(2)では直進移動、式(3)では衝突回避の適応度を計算している。1ステップ毎に累積し、計算の終了条件は4.2節の適応度計算と同じとする。

5.3 実験結果

100世代目でロボットは図4の環境の中を巡回する結果が得られた(図6参照)。目的が2つと少なく、環境が小さく単純なものであることから、学習が早く終了したと考えられる。この実験結果より、適応度関数を複数にしても、ロボットが学習を行えることを確認できた。

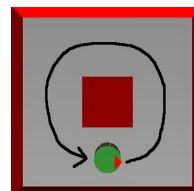


図6: 多目的の学習後の軌跡

6. 多目的の行動学習

次に、単目的では構成することができない、多目的から成る課題を取り上げる。

6.1 学習環境

ここでは「壁とロボットに衝突しないこと」と「目的地に行くこと」、この2つを目的とし、学習を行う。学習環境は図7で行い、目的地は右上のロボットが置いてある座標(2, -2)である。ロボットは全部で5台用意するが、そのうちの4台は4章で得られたコントローラを利用して動作させる。その他の、白い矢印で示されている1台のロボットのみを学習させる。ここで、4台のロボットが3章のコントローラを使用しているのは、シミュレータのセンシング情報取得の間隔による制約により、ランダムに動き続けていると学習ができないためである。

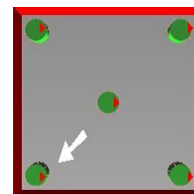


図7: 学習環境

6.2 適応度関数の設定

学習には、多目的遺伝的アルゴリズムのNSGA-IIを使用した。個体数60、交叉率0.9、突然変異率0.1とし、ここでの適応度関数は、以下の2つで設定した。

$$f_1 = \sum \{v + (\pi - |r|) + s\} \quad (4)$$

$$f_2 = \sum_t \{ \sqrt{(2-x_t)^2 + (-2-y_t)^2} - \sqrt{(2-x_{t-1})^2 + (-2-y_{t-1})^2} \} \quad (5)$$

f_1, f_2 : 適応度 v : 速度
 $|r|$: 回転速度の絶対値 s : センサ値の平均
 (x_t, y_t) : t ステップ時の位置座標

式 (4) は、あまり方向転換をせず、壁に衝突しない場合に適応度が高くなるようにし、式 (5) では、ステップ毎に目的地に近づく、適応度が高くなるように設定した。計算の終了条件は、1000 ステップまで計算が終了するか、ロボットの速度が 0、1 ステップ間の移動距離が 0、目的地に到達することである。また、壁やロボット同士で衝突した際には、どちらの関数も適応度を 0 にし、終了条件が満たされるまで計算を続ける。

6.3 学習結果

障害物となる 4 台のロボットの初期位置を固定した環境下で 200 世代まで学習を終えたコントローラは、障害物となるロボットを避けながら目的地へ到達し、先述した 2 つの目的を達成できたことが確認された。学習後に獲得されたコントローラを使用し、図 8 の環境で動作確認をした。ここでは、左下の $(-2, 2)$ のロボットに学習後のコントローラを適用し、他のロボット 4 台は 4 章で得たコントローラを適用し、初期位置はランダムに出現させた。図 8 に学習後のロボットの動作を示す。

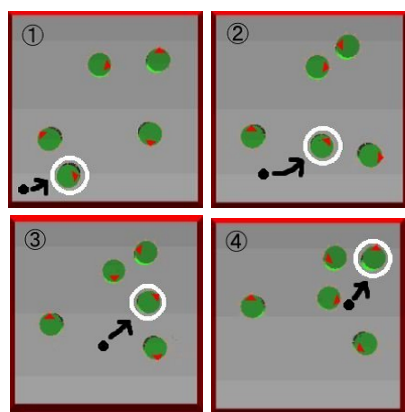


図 8: 学習後の動作

この結果より、学習環境とは違う環境 (ロボットの初期位置が異なる) でも動作可能であることがわかった。今回の学習を行う際に、障害物となるロボットの初期位置を毎回同じ場所から始めていることは、動的な環境を対象にする際に、毎回異なる環境をロボットが学習することは学習の効率が下がることが容易に考えられるため、固定された場所に配置した。学習後のロボットは、障害物となるロボットがランダムに出現しても衝突を回避して目的地へ向かう動作を行うことが確認され、結果的に効率の良い学習ができたと考えられる。

また、図 7 の環境において、適応度関数を 6.2 の 2 つの関数の和とし、単目的として学習を行った場合、学習後のロボットは、他のロボットを避けるように学習できているが、目的地へ到達することはできない。これは、2 つの目的を 1 つの適応度関数にしてしまったことで、何を達成すれば適応度が高くなるのかが不明瞭になってしまうためだと考えられる。

7. おわりに

本研究では、複数の問題を取り扱うロボットの行動学習の基礎的検討として、進化計算における多目的最適化手法として使用されている NSGA-II を用いたロボットの行動学習について実験と考察を行った。初めに、通常、単目的で解決可能な問題を、複数の目的に基づいて解決される問題に変更し、多目的最適化手法を用いてロボットが行動学習を行えることを確認

した。その後、単目的では解決不可能な問題を取り上げ、多目的最適化手法によって、そのような問題が解決できることを示した。

多目的最適化手法は、複数の目的を同時に満たすように学習を行う手法であり、そこに目的どうしのトレードオフが生じるため、複数の目的をすべて満足するように課題を解決する行動学習には向いていないと考える。今後は、目的ごとに問題を解決する学習経験を組み合わせることによって、複数の目的を達成するコントローラの実現を目指す。

参考文献

- [1] I.Harvey, P.Husbands, D.Cliff, Issues in Evolutionary Robotics, Proceedings of SAB92, the Second International Conference on Simulation of Adaptive Behaviour, In J.Meyer, H.Roitblat, S.Wilson, editors, MIT Press Bradford Books, Cambridge, MA, 1993
- [2] 三浦 正徳, 田口 英郎, 岡田 奈津子, 福島 充, ロボットアームのニューラル制御: 書字動作による検証, IEICE technical report, Neurocomputing 97(332), 45-50, 1997
- [3] 日高 良和, 久保田 智彦, 階層型ニューラルネットワークを用いた移動ロボットの障害物回避, 宇部工業高等専門学校研究報告 47, 19-25, 2001-03
- [4] 有江 浩明, ブルウィーラー ベアート, 尾形 哲也, 谷 淳, 菅野 重樹, ニューラルネットワークによる腱駆動ロボットアームの制御: ランダムな運動からの逆モデルの学習, ロボティクス・メカトロニクス講演会講演概要集 2007, "2A1-B10(1)"-"2A1-B10(2)", 2007
- [5] Diane Law, Risto Miikkulainen, Grounding Robotic Control with Genetic Neural Networks, Technical Report AI93-223, Department of Computer Sciences, The University of Texas at Austin, 1994
- [6] Dario Floreano, Francesco Mondada, Automatic Creation of an Autonomous Agent: Genetic Evolution of a Neural-Network Driven Robot, Proc. Of the Third International Conference on Simulation of Adaptive behavior: From Animals to Animats 3, The MIT Press/Bradford Book, 1994
- [7] Stefano Nolfi, Evolving non-Trivial Behaviors on Real Robots: a garbage collecting robot, Institute of Psychology, National Research Council, Technical Report 96-04, 1996
- [8] 仲間 祐貴, ラル スニル プラニット, 山田 孝治, 遠藤 聡志, ニューラルネットワークを用いたモジュール型ロボットの制御, ロボティクス・メカトロニクス講演会講演概要集 2008, "2A1-G24(1)"-"2A1-G24(2)", 2008
- [9] David J.Montana, Lawrence Davis, Training Feedforward Neural Networks Using Genetic Algorithms, Proc. 11th Joint Conf. Artificial Intelligence Vol.1, pp762-767, 1989
- [10] 高橋 裕樹, 中嶋 正之, 遺伝的アルゴリズムを用いた最適な階層型神経回路網の構造決定に関する一考察, 電子情報通信学会論文誌 D-II Vol.J79-D-II No.11, pp1920-1928, 1996
- [11] 片田 善章, 進化ロボティクスにおける多目的 GA に関する一考察, 第 54 回システム制御情報学会研究発表講演会, F34-1, 2010
- [12] 東原 智幸, 渥美 雅保, 多目的 GA を用いたロボットの行動決定ネットワークの進化的獲得, 情報処理学会第 111 回知能と複雑系研究会研究報告, 98-ICS-111, pp.1-6, 1998
- [13] <http://simbad.sourceforge.net/>
- [14] S.Nolfi, D.Floreano, Evolutionary Robotics, MIT Press, 2000
- [15] Louis Huges, Nicolas Bredeche, Simbad: an Autonomous Robot Simulation Package for Education and Research, Simulation of Adaptive Behavior, 2006
- [16] K.Deb, A.Pratap, S.Agarwal and T.Meyarivan, A Fast Elitist Multiobjective Genetic Algorithm: NSGA-II, IEEE Transactions on Evolutionary Computation, Vol.6, pp.182-197, 2000