2U-8

遺伝的アルゴリズムを用いた強化学習モジュール化手法

吉田　裕昭†　　中村　真吾¶　　橋本　周司‡

早稲田大学大学院 先進理工学研究科 物理及応用物理学専攻†

芝浦工業大学¶　早稲田大学 理工学術院‡

1. はじめに

強化学習[1]は，エージェントが環境との試行錯誤により行動の目標に応じた報酬の総和を最大にするような行動則を獲得するための枠組みである．しかし，状態数や行動数の多いシステムに対する最適な制御器を獲得しようとすると，状態空間が指数関数的に拡大し，膨大な学習時間が必要となる．この問題の解決策として複数の単純な制御器を用意し，系全体の制御方法を学習するモジュール型強化学習が提案されている[2][3]．しかしながら，個別の部分モジュールの入力として何を選ぶかは，人の手によって決定されている．このため，設計者が対象に関する予備知識を必要とする．また，本来学習に不要または過剰な状態を学習器に渡した場合，学習が収束しないといった問題がある．そこで，本研究ではモジュール型強化学習法の状態選択を遺伝的アルゴリズムで決定するモジュール型強化学習アルゴリズムを提案する．実験では，テレビゲームのキャラクタの操作に提案手法を適用し，環境適応性の向上を検証した．

2. 提案手法

2.1 モジュール型強化学習

まず，本研究におけるモジュール型強化学習について説明する．システムは制御目的に応じた複数の制御モジュールを持つ．m 番目のモジュールは対応した状態 $\boldsymbol{s}_m$ を観測し，行動 $\boldsymbol{a}_m$ を出力する．行動 $\boldsymbol{a}_m$ の k 番目の要素 $a_{m,k}$ はゲート G_k に渡され，ゲートの選択則に従ってどのモジュールの行動要素が選択されるか決定される．最終的に，各モジュールの行動要素を組み合わせた行動 $\boldsymbol{a}$ がシステム全体の制御として出力される．このとき各モジュールの報酬 r_m は制御目的に応じて決定される．図 1 にモジュール数 2，行動要素数 3 の場合のアルゴリズムの概要図を示す．

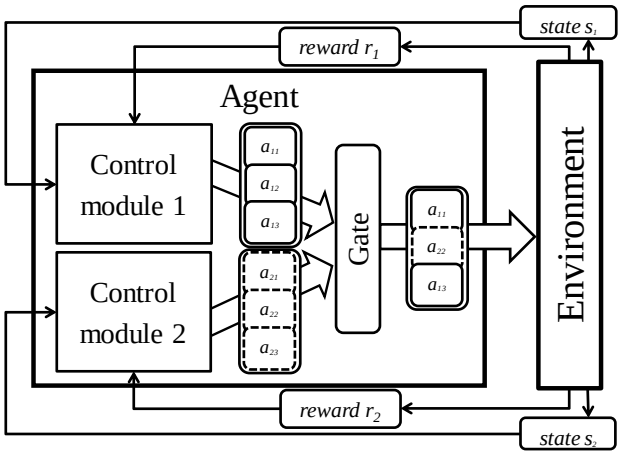

図 1　モジュール型強化学習

2.2 遺伝的アルゴリズムによる状態選択

本研究では，遺伝的アルゴリズムによりモジュール m が観測する状態 s_m を決定する．各モジュール m は，遺伝子情報に従って，システムの n'個の状態から n'個を選択し，s_m として状態を観測する．ここで，遺伝子はモジュールがどの部分状態を入力とするか，入力の状態選択を 0 と 1 で表現している．モジュールが観測する状態を遺伝子の 1，観測しない状態を 0 として，個体は次のようにコード化する．

(01011……0)　　(1)

したがって，コードの長さは全状態空間の次元数 n，1 の数は部分状態空間の次元数 n'と等しくなる．各モジュールはそれぞれ複数の個体を持ち，最も評価の高い個体からランキング選択方式によって次世代へ引き継がれる．

交叉では親の各遺伝子コードを加算し，結果が 0 の遺伝子は 0，2 は 1 とする．加算結果が 1 となった遺伝子は半数をランダムに 1 とし，残りを 0 とする (図.2 左)．突然変異では親遺伝子の 0 と 1 をランダムに入れ替える(図.2 右).

2.3 寄与率

各遺伝子を定量的に評価するため，新たな指標を定義する．モジュール m の個体 i が選択した行動が報酬を受け取る際に，どの程度貢献したかを測る寄与率 $K_{m,t,i}$ を以下のように定義する．

Approach to modularize the reinforcement learning that uses a genetic algorithm

†Hiroaki Yoshida, Department of Applied physics, Waseda University

¶Shingo Nakamura, College of Engineering, Shibaura Institute of Technology

‡Shuji Hashimoto, Faculty of Science and Engineering, Waseda University

$$K_{m,t,i} = \sum_{k=1}^{\infty} C_{t-k,i} \lambda^{k-1} \quad (2)$$

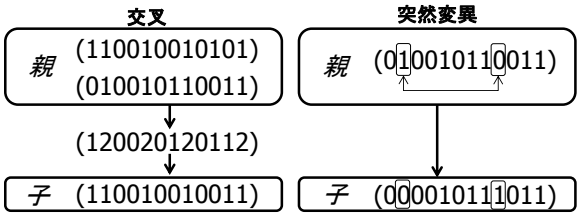

図 2　交叉と突然変異

ここで, t は報酬を受け取った時の時刻, $C_{t-k,i}$ は時刻 $t-k$ におけるモジュール m の個体 i 出力した行動 $\boldsymbol{a}_{m,i}$ と実際のエージェントの行動 $\boldsymbol{a}_i$ との内積である. λは 0<λ<1 を満たす寄与率の割引率である. この寄与率を遺伝的アルゴリズムの評価関数として使用する.

3.実験

3.1 実験対象

実験では，オープンソーステレビゲーム『INFINITE MARIO BROS』[4]に提案手法を適用し，性能を評価した．このゲームはスクロール型アクションゲームで，制御するキャラクタをステージ右端まで進めることを目的とする．ステージには所々に段差や落とし穴，敵等の障害物が設置されており，地形の状態の多さや，時間の経過により状態が刻々と様々に変化するといった点で複雑な制御を必要とする.

ゲームステージは制御キャラクタの幅の 150 倍とし，9 度グラウンドレベルが変化するものとした．更に，落とし穴を 2 つ設置し，踏み倒す以外に接触してはいけない敵キャラクタの数を 1 に設定した.

3.2 実験設定

実験ではモジュール数を 3 とし，各モジュールに与える報酬値を行動結果ごとに表 1 に示す. この実験で観測できる状態は，s_1 が敵の x, y 位置と移動方向，s_2 が穴の x, y 座標と幅，s_3 が段差の x, y 座標と高さとし，これらにキャラクタの状態（x,y 方向の速度，ジャンプ状態）を加え，全 12 種類とした．これらのうち，どの状態をモジュールが観測できるかは，個体の遺伝情報によって決定される．行動は，ゲームを制御するジャンプ・ダッシュ・方向の 3 種類のボタンの ON/OFF とし，全部で 8 種類とした．強化学習の Q 値更新は 3 フレームごとに 1 回とし，更新回数が 2 万回ごとにステージクリア率を求めた．その後強化学習時の評価によって遺伝子の世代交代を行い，5 個体から次世代をつくり再度強化学習をする．世代数は 100 世代までとした．学習率 α = 0.3, 割引率 γ = 0.9, τ = 20.0, λ=0.5, 個体数を 60 にし，学習を行った.

表 1　報酬定義

種類	行動結果	値
r_1	右に進む	1 ステップに進んだ距離
	壁に衝突	-20
r_2	落とし穴を飛び越える	+100
	落とし穴に落ちる	-100
r_3	敵を倒す	+100
	敵と接触	-100
	敵を飛び越える	+10

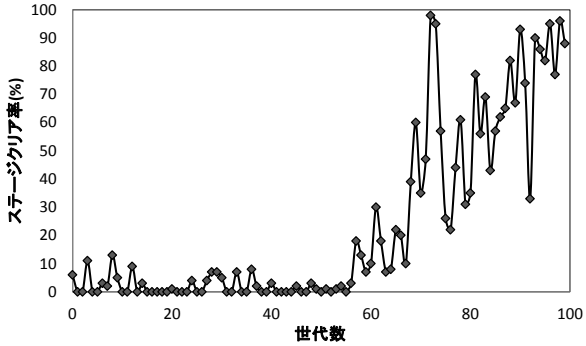

図 3　世代数とステージクリア率の関係

3.3 結果

各世代の強化学習後のステージクリア率を図 3 に示す．約 70 世代ほどでステージクリア率が急激に上昇するといった結果が得られたが，これは課題達成に必要な状態を各モジュールが観測したことによって，学習成功率が飛躍的に上昇したと推察できる．また，ステージクリア率が減少する世代では，強化学習がうまくいかずに局所最適解に陥ってしまったものと考えられる.

4. まとめ

モジュールが観測する状態選択を自動で行う，遺伝的アルゴリズムを用いたモジュール型強化学習法を提案した．また，これをテレビゲームのキャラクタ制御に適用することで有効性を確かめた．今後はモジュールの数を自動で調整可能なアルゴリズムに発展させたいと考えている.

謝辞
本研究の一部は，早稲田大学ヒューマノイド研究所，グローバルCOE プログラム「グローバル ロボット アカデミア」の研究助成を受けて行われた.

参考文献

[1]Richard S.Sutton and Andrew G.Barto, “Reinforcement Learning; An Introduction”, The MIT Press, 1998.
[2]山田訓, “モジュール型強化学習”,信学技報, NC97(623), pp.139-146, 1998.
[3]中間隼人ら, “3 種類のセンサを持つロボット制御へのモジュール型強化学習の適用”, 電子情報通信学会, NC108(480), pp.301-306, 2009.
[4]M.persson, “INFINITE MARIO BROS” Available: http://www.mojang.com/notch/mario/