Vol.2016-MPS-110 No.10
2016/9/16

遺伝的アルゴリズムを用いた QWOP の運動学習の一手法

小山ちひろ† 石川由羽† 高田雅美† 城和貴†

概要：本稿では，ルールや攻略法を与えず自動的にルールを構築し，上達することができる成長型ゲーム AI の開発を行う．専用の学習エミュレータが存在しないゲームを学習させることは困難である．そこで，プレイ画面情報のみでゲームを学習することができるシステムを開発する．画面情報より得られたスコアのみを学習に用い，遺伝的アルゴリズムを適用して Flash ゲーム“QWOP”の運動学習を行う．ランダムな行動パターンを持つ個体群に対し，遺伝子操作を行うことでスコアが徐々に向上し，歩行運動を学習していくことを確認する．

Movement Learning for QWOP using a Genetic Algorithm

CHIHIRO KOYAMA† YU ISHIKAWA†
MASAMI TAKATA† KAZUKI JOE†

1. はじめに

ゲームは，対戦型，シミュレーション型，アクション型など多くのジャンルに溢れ，制作技術は日々進化している．ゲームにおける人工知能（Artificial Intelligence, AI）分野ではチェスや囲碁，将棋，オセロなどのボードゲームで人間に勝つことができるゲーム AI が開発されている[1]．これらのゲーム AI は開発の際，基本的なルールや仕組み，思考ルーチンを与えられている．ルールは，攻略としてまとめられている場合や，対戦履歴を蓄積することによってパターン化されている場合がある．しかしながら，これらはユーザが生み出した攻略法であり，必ずしも，高スコア獲得のための最適な手段であるとは限らない．また，新しいゲームやユーザが少ないゲームでは，攻略法が十分ではない可能性がある．このような場合，ゲーム AI がユーザと同様にプレイすることで確実な攻略法を得られると考えられる．そのために，ルールを与えずにランダム操作でプレイさせたデータを学習に用いることで，徐々にスコアの獲得方法を会得し，自動でルールを構築することができる成長型ゲーム AI の開発が望まれている．既存の手法[2]では，遺伝的アルゴリズムを用いて教師無し学習させることで，攻略に関する知識が無い状態から徐々に上達し，人間を超えるスコアを獲得することが可能となっている．遺伝的アルゴリズムとは生物が環境に適応して進化していく過程を工学的に模倣した学習アルゴリズムである．この学習アルゴリズムをゲームに適用させることで，世代を重ねれば重ねるほど良い行動パターンが得られ、人間では考え難いようなプレイでゲームを攻略できる可能性がある．このような成長型ゲーム AI の既存研究では，レトロゲームやボードゲームなど，操作がシンプル，かつゲーム進行が容易なゲームを対象にしている．そこで本稿では，操作が難しく，攻略するための方法が幅広く存在する自由度の高いゲームを対象とする．その中でも，ブラウザ上で誰でも無料でプレイすることができる FLASH ゲームを対象とする．また本稿では，ゲームから取得できる情報がプレイ画面のみである．そのため，1 種類の情報でも評価用のデータとして用いることで学習することができる遺伝的アルゴリズムを適用する．

以下，2 章で本稿の対象とするゲームの仕様について紹介する．3 章では新たな学習システムの手法を提案する．提案する学習法には，遺伝的アルゴリズムを適用する．4 章では実験環境と学習システムを用いた実験結果と考察を述べ，5 章でまとめを述べる．

2. QWOP

本稿では，QWOP と呼ばれる FLASH ゲームを対象とする[3]．図 1 は QWOP のプレイ画面とゲームオーバー時の画面である．このゲームは，キーボードの “Q”, “W”, “O”, “P” をタイミング良く入力することで画面内の走者を走らせ，100m まで到達することでクリアとなる．また，膝が地面につくことによりゲームオーバーとなる．各操作の動作は，以下の通りである．

・Q … 左腿を後ろに引く
・W … 右腿を後ろに引く
・O … 左ふくらはぎを後ろに蹴る
・P … 右ふくらはぎを後ろに蹴る
・R … スタート地点へリセット

QWOP では，進んだ距離をスコアとして，画面の上中央

† 奈良女子大学
Nara Women’s University

図 1　QWOP のプレイ画面(上)とゲームオーバー画面(下)

部に表示している．スコアは進む距離によって変動する値であるため，このスコアを用いて学習を行う．このゲームは操作キーが少なく，簡素であるが，物理エンジンが働かせる重力により，キーを入力するタイミングが非常に厳正となっている．また，スタートする際に，走者が左右に揺れており，適切なタイミングで第一歩を踏み出す必要がある．

既存の手法では専用のエミュレータを用いているが，本稿で対象とする FLASH ゲームには存在しない. そのため，ゲームの学習をするために，最低限必要なゲーム操作の自動化を行う．本提案手法ではシェルスクリプトを用いて GUI を操作し，キー入力操作を自動で行う．また，プレイ画面全体の情報を用いるのではなく, 一部分のみを利用し，画像認識により必要な情報を得る．

3. 運動学習法の提案

本稿では，結果のスコア情報を用いるシステム開発を行う．この際，スコア情報のみで学習が可能な遺伝的アルゴリズムによる探索によって解の取得を試みる．

3.1 遺伝子コーディング

QWOP への入力は，キー入力の 1 次元配列として捉えることができる．そのため，この 1 次元配列を最適化する遺伝的アルゴリズムを適用する．詳細については，3.2 節で述べる．また，個体の遺伝子情報は，プレイ 1 回分の行動パターンとする．QWOP は単独のキー操作だけでなく同時押しにも対応しているため，キーを組み合わせた操作も行動の 1 つとして加える．ゆえに，1 個体が持つ遺伝子は，

表 1　ランダムな行動パターンの例

	行動 1	行動 2	行動 3	…
個体 1	Q	QP	O	…
個体 2	QP	Q	W	…
…	W	WO	WO	…

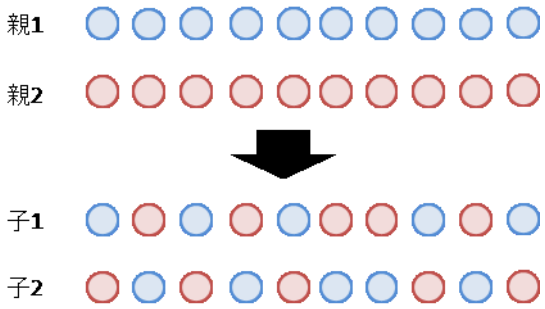

図 2　一様交叉

“Q”，“W”，“O”，“P”，“Q+P”，“W+O”，“none”の各入力時間で表される行動パターンとなる．

3.2 遺伝的アルゴリズムの手順

本節では，遺伝的アルゴリズムの手順を説明する．画像認識によりスコアを取得し，適応度として個体に与えることで，より優秀な個体を選出する．本稿での各個体への評価は，プレイ中に進んだ距離をそのまま適応度とする．

手順 1. 個体生成
手順 2. 適応度計算
手順 3. 親の選択
手順 4. 遺伝子の交叉
手順 5. 遺伝子の突然変異

手順 1.では，初期値として表 1 に示すようなランダムな遺伝子(入力時間別キー操作)による行動パターンを持った個体を複数用意する．

手順 2.では, 用意した個体を実際に 1 個体ずつ実行させ，進んだスコアを適応度として個体に与える．遺伝子 1 つ分の入力を行う毎にスコアを取得し，変動を見る．スコア取得の詳細は 3.3 節で述べる．また，1 個体の実行後に次の個体の実行へ移る際，”R”キーでリセットを行うと前個体が直前まで実行していたキーが押されたままの状態になるなど，次個体の実行に影響を受けてしまう場合がある．そのため，1 個体の実行終了後，リセットとしてブラウザを更新することで回避する．

手順 3.では，個体群から複数個体を選択し，その中から適応度が高い 2 個体を選ぶ．本稿ではトーナメント方式を用いる．個体群から複数の個体をランダムに選び，その中から最も適応度が高い 2 個体を親とする．

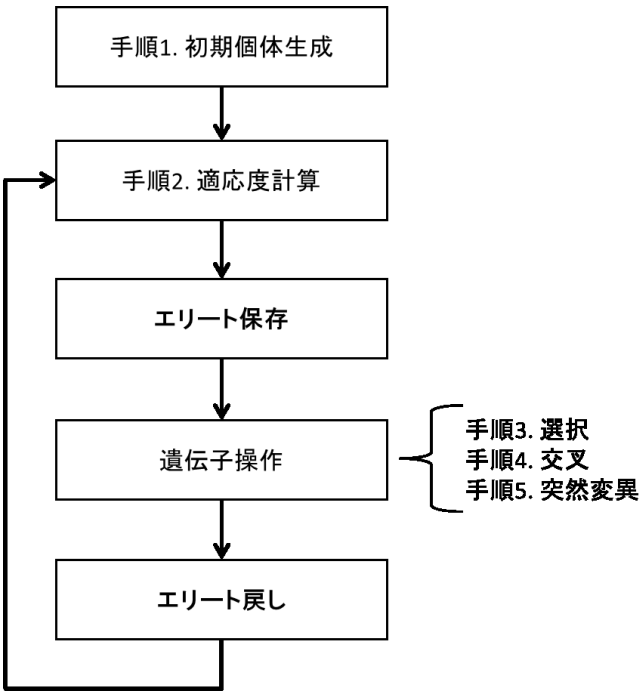

図 3　エリート保存とエリート戻しの順番

手順 4.では，手順 3.で選んだ 2 個体の親個体の遺伝子を一定の確率で組み換え，子個体を作り出す．図 2 は，本稿で用いる一様交叉のイメージ図である．一点交叉や二点交叉は，親 2 個体の遺伝子を共通の箇所で切断して組み替える方法であり，組み替えのバリエーションが少ない．一方，一様交叉は，遺伝子を 1 つずつ組み替える方法であり，生成される子のバリエーションが多い．また，本提案手法では遺伝子の数が多いため，一点交叉や二点交叉の場合，学習の効率が悪いと予想される．よって，一様交叉を適用する．以降，子個体の数が初期個体数と同じ数になるまで選択，交叉を繰り返す．この個体群が次の世代となる．

手順 5.では，次世代の個体群の中で 1 つの遺伝子を一定の確率で変異させる．これは局所最適解に陥ることを防ぐためである．

以上のように次世代を生成し，以降，これらを親世代として手順 2.から手順 5.までを設定した世代の数だけ繰り返す．また，適応度の高い個体が生成されても遺伝子操作を行うことによって次世代に残らないことがある．このことを避けるために，エリート保存戦略と呼ばれる手法が提案されている．これは，個体群の中で最も適応度の高い個体をそのまま次世代に残すという手法である．エリート保存戦略を適用する場合のエリート保存とエリート戻しの順番を図 3 に示す．エリート保存は手順 2.にて全個体への評価が終了した後に行い，現在の個体群から適応度の高いエリート個体を 1 個体選ぶ．エリート戻しは手順 3.から手順 5.にて遺伝子操作を行った後に行い，現在の個体群から適応度の低い個体を 1 個体選び，エリート個体を上書きする．

図 4　QWOP の数字フォント

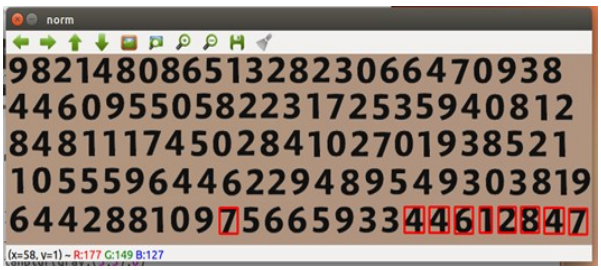

図 5　学習の様子

3.3　スコア取得方法

開発するシステムでは，プレイ画面上のスコア表示部分から画像認識で数字を認識して値を取得する．数字を認識する手段として，QWOP で使用されている数字のフォントをニューラルネットワークで学習させることで数字の認識を行う．

数字認識を行う手順を説明する．

手順 2-1. 数字フォント抽出
手順 2-2. 学習用データセット作成
手順 2-3. ニューラルネット学習
手順 2-4. スコア認識

手順 2-1.では，QWOP を逆コンパイルし，ゲーム内で使用されている数字画像を抽出する．抽出した数字画像を図 4 に示す．図 4 より，取得された数字は，すべて 1 桁の数字である．また，数字以外の小数点やアルファベットに関する情報は取得されていない．

手順 2-2.では，手順 2-1.で抽出した数字画像を用いて学習用データセットを作成する．図 5 は，学習用データセットを表示させた例である．学習用データセットは，各数字をランダムに複数並べた画像とする．

手順 2-3.では，学習用スクリプト[4]を実行して学習を行う．手順 2-2.で作成した学習用データセットを入力として学習用スクリプトを実行すると，黒線の面積を検知して輪郭の情報が取得される．その結果，図 5 のように 1 つの数字が赤枠で囲われる．囲われた数字をキーボードから入力すると，別の数字が赤枠で囲われる．この操作を続けることで数字画像と入力データを関連付ける．全ての数字画像に対する入力が完了すると，各数字の特徴量が入った学習済みニューラルネットワークモデルが構築される．

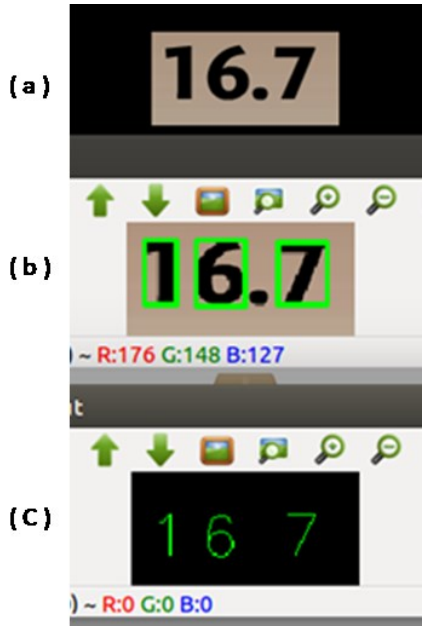

図 6　認識結果

手順 2-4.では，学習済みモデルを用いて数字認識を行う．まず，プレイ画面上の距離表示部分のみをスクリーンショットで取得する．この時，取得する範囲は手動で設定する．また，数字の桁数によってスクリーンショットを撮る範囲から読み取りたい数字が外れてしまう場合があるため，数字だけでなく“metres”を含めた範囲を撮る必要がある．取得された画像内の数字の線は白に近い色であるため，背景と区別しにくい．そこで，撮影した画像に色調反転処理を施して線を黒くする．次に，この画像から数字を認識する．認識させた結果画像を図 6 に示す．図 6(a)は距離表示部分の画像を色調反転処理を施した状態，(b)は(a)から数字として認識できる部分を枠線で囲った状態，(c)は囲われた部分から認識した数字を認識した位置に表示した状態である．手順 2-3.で構築された学習済みモデルは，1 桁の数字のみを認識することができる．そのため，図 6 の状態では，数字を 1 桁ずつ認識して表示している．つまり，各数字の位，小数点やマイナス記号は認識されていない．しかしながら，スコアを正確に取得するためには，これらの情報が必要となる．そこで，小数点やマイナス，数字の位を含めて認識させるため，認識した数字の位を割り振る必要がある．図 6 のような結果に小数点やマイナス記号に関する情報を含ませるために，認識した 1 桁の数字の位を割り振る処理を行う．そのためのフローチャートを図 7 に示す．撮影した画像の XY 軸を図 8 に示す．“metres”を数字と誤認識することを防ぐため，“m”の表示部分の X 座標を取得する．この位置から X 座標が降順になるように位の低い数字から認識し，順に配列に代入する．この時，認識した数字の X 座標に関する情報をメタデータとして保持する．　また，走者がスタート位置より後ろに下がった場合，スコア表示部分は図 9 の上図のように表示される．このマイナス

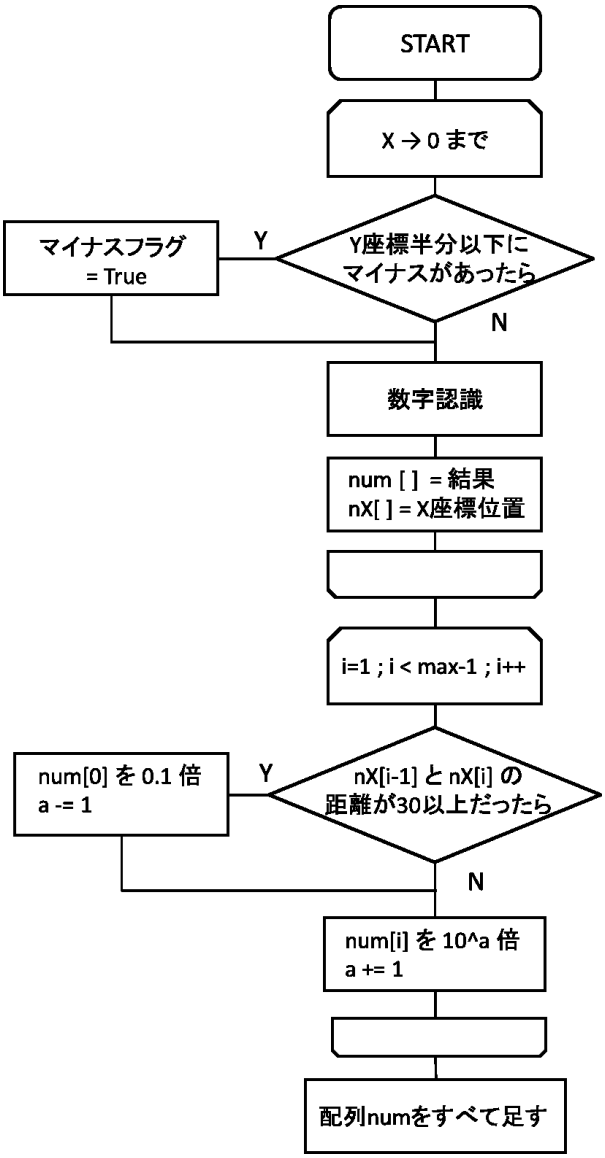

図 7　位割り振り手順

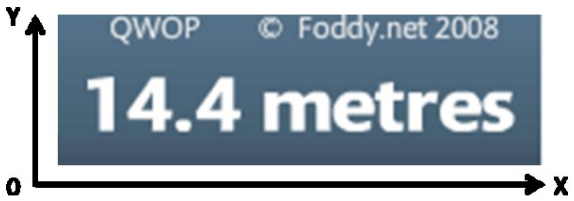

図 8　スコア画像の XY 軸

を検知するため，XY 座標が最も小さい位置に面積の狭い，黒いピクセルが認識された場合はマイナスフラグを正にする．全ての数字が配列に代入された後，小数部の判定を行う．小数点は最も面積が狭く，低い位置に表示されるため，マイナスと誤認識されやすい．そのため，各数字の間隔によって小数部の判定を行う．小数部と整数部の間隔は整数同士の間隔より離れているため，配列の隣同士の要素と X 座標の位置関係から判定することができる．小数部と判定

図 9　マイナス表示(上)と位割り振り成功画面(下)

されるとその数字を 0.1 倍し，以降の数字を 1 倍，10 倍する．最後にマイナスフラグが正の場合，-1 倍する．以上の手順で図 9 の下図のように正確な値を取得することができる．

4. 実験

4.1 実験方法

本稿では，歩行運動を学習することを目標に，大きく分けて 3 つの方法で実験を行う．また，本システムではゲームオーバーになってしまった場合の判定を行うのが困難であるため，替わりに目標条件を設定する．1 個体の実行中に目標条件を達成できなかった場合，その時点での進んでいる距離を適応度として付加し評価を終了する．

まず方法 1 として，1 個体の行動中，目標条件を変化させることで，学習にどのような影響が出るか試みる．目標条件(a)は 3 秒間スコアが停止するまでとした場合，(b)は 1 秒間スコアが停止するまで，かつ 5 秒で 3.0m 未満を目標条件とした場合とする．また，個体の遺伝子数を 300 とし，個体数は 100 個体，交叉率は 0.8，突然変異率を 0.1 とする．遺伝子の種類は“Q”，“W”，“O”，“P”，“Q+P”，“W+O”，“none”の入力時間各 0.1 秒分を表す全 7 種類とする．

次に方法 2 として，個体数を 200 個体に増やし，エリート保存戦略を適用する．目標条件を 1 秒間スコアが停止するまで，かつ 5 秒間でスコア 1.0m と設定する．また，入力時間のバリエーションを考え遺伝子の種類を増やす．著者がプレイした場合の入力データを参考に，使用頻度の高い“O”，“P”，“Q+P”，“W+O”，“none”の 0.05 秒分，0.25 秒分，0.5 秒分の遺伝子と，バランス調整の為少々使用される“Q”，“W”の 0.05 秒分の遺伝子の全 17 種類とする．

最後に方法 3 として，エリート保存戦略を適用した上で初期個体群の状態が及ぼす影響を確認する．そのために，初期個体群が完全ランダムの場合と，著者が実際にプレイしたデータを含ませた場合の実験結果の比較を行う．プレイデータの詳細は，著者が試用として 50m まで走行した記録である．このプレイ記録を遺伝子に変換し，初期個体群に 3 個体分含ませる．目標条件は方法 2 と同様とする．また，各遺伝子の入力時間を方法 2 の状態より長く設定する．

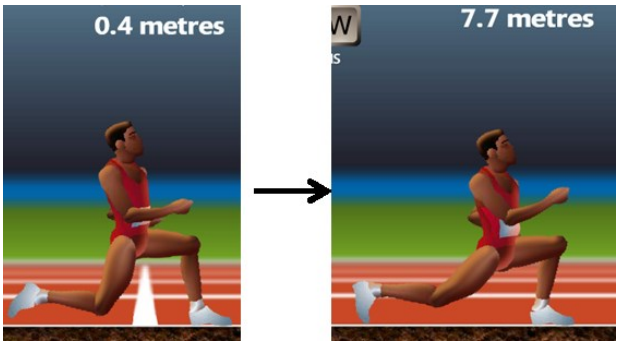

図 10　方法 1(a)での体勢

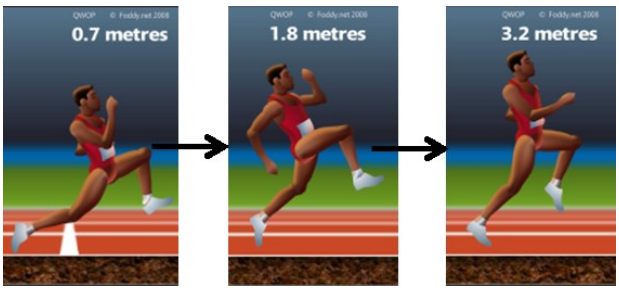

図 11　方法 2 での状態遷移

“O”，“P”，“Q+P”，“W+O”，“none”の 0.1 秒分，0.5 秒分，0.8 秒分の遺伝子と，“Q”，“W”の 0.1 秒分の遺伝子の全 17 種類とする．

4.2 実験結果

方法 1 での結果を表 2 に示す．(a)では，30 世代で最大 13.3m を観測している．また，行動の様子を図 10 に示す．右図はスタート直後の体勢である．前進運動を学習できているが，低姿勢のままを保ち，左図のように常に大股開きで徐々に進んでおり，両脚は交互に動いていない．一方(b)では，300 世代で最大 2.9m を観測している．こちらはスタート直後にすぐに転倒する．また，世代が進むにつれ，最初の 1 歩目までは順調だが，2 歩目が踏み出せずに転倒する．つまり，学習によって，徐々に最初の 1 歩目で助走をつけ，そのまま勢いよく前方へ倒れるようになっている．この様子から，(a)と(b)どちらも歩行運動は学習できていないことが確認できる．

方法 2 での結果を表 3 に示す．200 世代まで繰り返した結果，最大 9.3m を観測している．また，エリート保存数を複数に増やした結果，5 個体で 11.8m，10 個体で 8.2m を観測している．実験中の状態遷移を図 11 に示す．両脚を交互に細かく，あるいは大股で動かしており，歩行運動を学習できていることが確認できる．また，エリート保存数が 5 個体の場合に最も距離が長く，10 個体の場合では 1 個体の場合より短くなっている．

方法 3 での結果を表 4 に示す．エリート保存数を 7 個体とし 200 世代まで繰り返した結果，完全ランダムの初期個体群の場合では最大 12.1m，プレイデータを 3 個体含ませ

表 2　方法 1 の実験結果

目標条件	世代数	最長距離
(a)	30	13.3m
(b)	300	2.9m

表 3　方法 2 の実験結果

エリート保存数	最長距離(200 世代)
1	9.3m
5	11.8m
10	8.2m

表 4　方法 3 の実験結果

初期状態	最長距離(200 世代)	計算時間
完全ランダム	11.8m	約 82 時間
プレイデータ含む	13.6m	約 100 時間

た初期個体群の場合では最大 13.6m のスコアを獲得している．また，前者の計算時間は 82 時間，後者の計算時間は約 100 時間である．

4.3　考察

方法 1 の結果から，歩行運動が学習できない原因を考察する．(a)では，3 秒間スコアが停止している状況になる場合が少なく，どんな体勢であっても，より良いスコアを獲得するための攻略法を取得したと考えられる．この攻略法は，QWOP の攻略サイト[5]で紹介されている方法である．つまり，世代数を増やすことで，限られた時間内でより高いスコアを獲得するための攻略法を獲得したことを意味する．ゆえに，遺伝的アルゴリズムを用いて成長型 AI を開発することは有効であると考えられる．一方(b)では，時間毎に獲得させたいスコアを指定したことで前進ができなくなっている．この原因を考察すると，このゲームの仕様で，距離判定が走者の頭部についているため，より遠くの位置に頭部のみを到達させることが適切な解であると学習しているものと考えられる．また，歩行する為には細かいバランス調整やリズムカルな入力が必要な中で遺伝子 1 つの入力時間を 0.1 秒間に制限していることも，歩行運動が学習できなかった原因であると考えられる．

方法 2 の結果では，エリート保存戦略を適用させ，個体数と入力時間のバリエーションを増やしたことで，歩行運動を学習することができている．また，エリート個体を複数増やすことで距離が伸びる場合と縮む場合がある．つまり，全体個体数とエリート保存数には適切な割合があると考えられる．

方法 3 ではさらに歩行距離を伸ばすため，著者のプレイデータを初期個体群に用いた結果，完全ランダムの場合と大きな差は観測できていない．また，プレイデータを含ませた場合の方が計算時間が長い．つまり，各世代の中で長時間生き残る個体が多かったと考えられる．よって，少なからずプレイデータの影響を受けているものだと考えられる．また，この実験では一様交叉法を適用しているが，一点交叉法など親個体の遺伝子パターンを一部そのまま受け継ぐことができる交叉法を適用することで，プレイデータから受ける影響を拡大できると考えられる．

5. まとめ

本稿では，遺伝的アルゴリズムを用いた QWOP の運動学習法を提案した．QWOP を含む Flash ゲームには手軽に画面情報やスコア情報などの内部情報を取得できる専用エミュレータが存在しない．そこで，スコア表示部分を画像認識することによりスコアを取得し，適応度として用いている．最初に，1 個体のプレイを終了する条件を変化させることで学習にどのような影響が出るかを試みた．次に，エリート保存戦略を適用させ，個体数と遺伝子の種類を増やして実験を行った．最後に，さらに最長距離を伸ばすための方法として，初期個体群の遺伝子が完全ランダムな場合と，自分のプレイデータが入った個体を複数含ませた場合との比較を試みた．結果，歩行運動を学習していることが確認でき，歩行している状態で 10m 前後まで前進することが確認できた．この結果から，QWOP から得られる情報がスコアのみでも，遺伝的アルゴリズムを適用させることによって，進む距離を伸ばせることが確認できた．また，初期個体群の遺伝子がランダムの場合でも前進できていることから，教師データが用意できない場合でも，世代数を増やすことでより高いスコアが得られると考えられる．しかしながら，本稿の結果では，歩行している状態で 10m 前後まで前進することを学習するまでに約 4 日分の時間が掛かっている．世代数を増やすことに比例して計算時間も増加すると考えられる．そのため，今後は，歩行運動の学習をより早くするため，適切な遺伝子操作と目標条件を再考する必要がある．

参考文献

[1] 瀧澤 武信．人間に勝つコンピュータ将棋の作り方．技術評論社, 2012
[2] “Watch this learning AI smash Super Mario World with ease”. http://thenextweb.com/insider/2015/06/14/watch-this-learning-neural-network-annihilate-super-mario-world-with-ease/, (参照 2016-08-11).
[3] “QWOP”. https://www.foddy.net/Athletics.html, (参照 2016-08-11).
[4] ”Simple Digit Recognition OCR in OpenCV-Python”. http://stackoverflow.com/questions/9413216/simple-digit-recognitire-ocr-in-opencv-python, (参照:2016-02-15)
[5] “qwop の攻略法。”. http://tagblue.net/webservice/qwop-method.html, (参照:2016-08-11)