

2002年度卒業論文

遺伝的アルゴリズム（GA）を利用した
魚モデルの経路制御の最適化

指導教員：渡辺 大地

メディア学部3DCGアプリケーション構築プロジェクト

学籍番号 99p082

大草 むつみ

2003年3月

2002年度 卒業論文概要

論文題目

遺伝的アルゴリズム (GA) を利用した
魚モデルの経路制御の最適化

メディア学部
学籍番号: 99p082

氏名

大草 むつみ

主査

渡辺 大地

副査

和田 篤

キーワード

遺伝的アルゴリズム (GA) AI エージェントモデル 衝突判定 経路制御

現在エージェントモデルが様々な分野で利用されているが、これら動かす際には制御情報としてたくさんの数値 (パラメータ) を設定する必要がある。これらはその時々環境情報にそって設定する必要があり、その設定は手間がかかる作業である。

この研究では、上下左右方向の空間への移動が可能なエージェントモデル「魚モデル」に焦点を当て、各パラメータの最適化を AI の一種である遺伝的アルゴリズム (GA) を使う事で自動化し、アニメーションの経路制御を行う。

手法としては、あらかじめ障害物回避の方法をパターン化しておき、大きさ・最高速度・障害物判定の視界・回避行動に必要な数値などを含む遺伝子を GA により設定する。これを元に、移動経路を一定描画回数分計算する。壁、障害物との衝突判定の結果を各遺伝子の成績とし、成績を元に GA によって最適化する事で、毎回設定された状況に応じて 3DCG 空間上を障害物に当たらずに動く経路制御を実現した。

目次

1 . 序論

1 . 1 研究の背景と目的

1 . 2 研究概要

2 . 様々な人工知能 (A I) とその特徴

3 . 遺伝的アルゴリズム (G A)

3 . 1 基本的な仕組み

3 . 2 世代交代の手法

3 . 3 特徴

3 . 4 これまでに成された研究と応用

4 . G A を用いた魚モデルの経路制御

4 . 1 エージェントモデルにおける G A の適用

4 . 2 魚モデルとその経路制御パラメータ

4 . 3 G A による経路制御パラメータの最適化

4 . 4 衝突判定

4 . 5 視界による障害物判定と回避行動

4 . 6 魚モデルの遺伝子の設定と画面描画

5 . 結果

6 . 考察

7 . 謝辞

8 . リファレンス

1 . 序論

1 . 1 研究の背景と目的

現在、映画・ゲーム・アニメ・ロボットなど様々な分野で生物の動作を真似て動くモデルが作られている。ユーザーの望む動きを自動的に行ってくれるモデルを「エージェントモデル」と言い、これらを動かす際にはたくさんの数値（パラメータ）の調整が必要となってくる。例えば3DCGのアニメーションモデルなら、モデルの大きさ・色・形などの基本的な物体の情報の他に、基本的な動作の設定、画面上の一定の範囲にとどまる事・他の物体と衝突しない事を目的とした動きの制御情報などが必要となってくる。これらのパラメータを毎度手作業で調整するのは手間がかかる作業である。描画される環境の条件 空間の広さ・障害物の大きさ・場所など が変わった場合は再設定が必要になる場合もある。さらに複数の個体を描画する場合には数種類のパラメータの組み合わせが必要となる。

現在、魚モデル経路制御としては「バーチャルフィッシュ」という手法が存在する。この手法では実際の魚が泳ぐ姿を二方向から経路データを取りこみ、CG空間を泳がせる魚に適用する事で、視覚的にも動きが自然である魚の動きを描画している[1]。エージェントのパラメータ設定へのGA応用例としては、TVゲームへの応用がある[2]。この例ではゲーム内の要素の一つとして敵エージェントモデルのパラメータ強化が行なわれる際にGAが使用されている。しかし、3DCGアニメーションの経路制御の調整は手間がかかる作業でありながら、自動化する手法、特にGAを適用した例は研究としては提案されていなかった。

この研究はエンターテイメント分野へのGAの応用の一つとして、エージェントモデルにおけるパラメータ調整を人工知能(AI)の一つ、遺伝的アルゴリズム(GA)を使う事で自動化する。GAは、「評価関数のみで最適解が得られる」、「乱数の多用により結果が毎回異なる」という特徴を持つ。この特徴を生かし、GAによって環境に適応した、毎度違ったパラメータを作成し、障害物に当たらないアニメーションの経路制御を行う事が目的である。

1 . 2 研究概要

この研究では、3次元空間内を障害物を回避しながら動く魚モデルの経路制御パラメータをGAによって自動生成する事を実現した。魚モデルの動き方は、通常は直進し、障害

物が視界に入ったときに回避行動を取るものとした。モデルの大きさ・最高速度・障害物判定の視界・回避する際の回転角度などのパラメータを遺伝子とし、これをGAによって淘汰、世代交代させることにより最適化した。遺伝子の評価基準は障害物との衝突回数を採用した。

なお、この研究では3DCG空間上でアニメーションさせるにあたり、魚モデルが上下左右方向の空間への移動が可能なエージェントモデルである事から魚モデルの描画に焦点を置いた。

GAによってパラメータの調整を行った結果、衝突判定で遺伝子の良し悪しを判断するだけで、障害に衝突しない「魚モデルの経路制御」が行われた。図1はGAで生み出されたパラメータで経路制御した魚モデルが、通った軌跡を球で描画したものである。生み出されるパラメータは泳がせる空間を全く同じ環境にしても、「GAのランダム性」によって毎回全く違った。また、環境に変化を与えても、それぞれ適応する結果が出た。

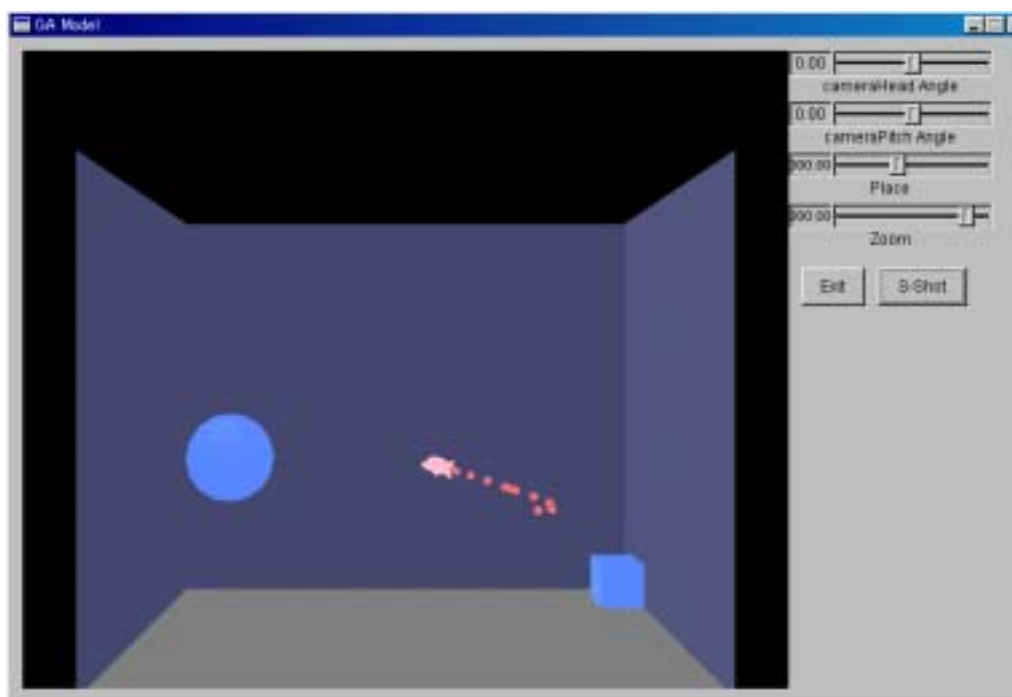


図1 魚モデルとGAによって経路制御された軌道（赤い球は魚の通った軌道）

次の2章では本研究のGAの適用理由として、現在主に使われている様々なAIを、しくみ・特徴・応用分野をあげて比較する。3章の1・2ではまずこの研究の基礎となるG

Aの一般的な仕組みと手法を詳しく解説し、さらに3・4ではGAの特徴を明らかにして具体的な研究・応用例を示す。4章では魚モデルの経路制御を、GAを使用して実現した方法について具体的に述べる。5章ではGAが出したパラメータの結果を中心として、この研究全体の結果を、6章では考察を述べこの研究のまとめとする。

2．様々な人工知能（AI）とその特徴

AIとは人間の知的活動を機械に行わせるという目的で始まった研究分野とその成果であり、その取り組み方には二つの立場がある。一つは人間の知能そのものをもつ機械を作ろうとする立場、もう一つは人間が知能を使って行なうことを擬似的に機械にさせようとする立場である。本研究は後者の立場に属する。AIの手法には様々な種類がある [3]。それぞれテレビゲーム・スケジュール設定・天気予報・ロボットの思考・インターネットの検索技術などエンターテイメントから研究まで実に幅広い分野で使われている。

この研究で使用したGAは遺伝と進化の仕組みを用いて解答を見つけ出すアルゴリズムであり、「ニューラルネットワーク」や「エキスパートシステム」と並ぶ代表的なAIの一つである。今回の研究では、他のAIと特徴を比べた結果パラメータの設定にGAを採用した。

この章では様々なAIとその特徴について述べる。

・遺伝的アルゴリズム（GA）

「遺伝子と進化」の仕組みを元にした進化型AI。様々な組み合わせの中から最適な組み合わせを見つける事が出来る。

関連性のある情報を一つながりの遺伝子として扱う。複数の遺伝子を乱数によって用意する。遺伝子の中から優秀なものを選び出し、実際の進化のように交叉と突然変異を繰り返して、劣等な遺伝子を淘汰し、優秀なものを生み出す。途中経過で新しく生み出される遺伝子が必ずしも以前のものより優秀とは言えないが、何度も繰り返す事で目的に合った最適解に近づけていく事が可能である。

評価基準の操作によって条件の変化に容易に応じる事が可能な事が長所である。乱数を多く利用する為に毎回選ばれる解答が同一に定まらない点がGAの出す解答の特徴である。

・遺伝的プログラミング (GP)

GAの応用で、グラフ構造(木構造)で扱えるように遺伝子の型を拡張する事で、プログラム生成や学習、推論、概念形成などに応用するアルゴリズム[4]。GPの考え方をを用いて、学習、推論、問題解決を実現する試みを進化論的学習と言う。

応用例としては株式データの予測[5]・ロボットのプログラミングなどがある。

・ニューラルネットワーク (NN)

「神経回路」の仕組みを元にしていく例題学習・推論型AI。脳の細胞とシナプスのように、様々な要素とそのつながり度合いの情報を保持し、例題の学習によってつながり度合いを変化させていく。学習させた後に何らかの情報をインプットすると今までの情報から推理して予測や判断を出す。

例題の学習が必要ではあるが、その後は未知の情報に対しても推論が可能である事が長所である。NNを使用した音声認識技術[6]などが研究されている。

・強化学習法

独自で学習する能力を持つ試行錯誤型AI。たくさんの選択肢がある場面で試行錯誤を繰り返し、行動を評価していく事で正しい選択に近づけていく。一つの場面から成否の成果を出すのではなく、一連の流れの中で成否を出しそれを元に学習する堅実なAIである。

高度の推論、判断、記憶などがなくとも試行錯誤をさせる事さえ可能であれば、問題を解決できる事が長所であり。1990年代から研究が始まり将来性を期待されている。

・エキスパートシステム

「考え方」を元にした知識データベース型AI。自分で学習する能力がないので専門家によって「場合」「対処法」「判断」「予想」などのルールを用意して蓄積しておく。条件や状態を入力するとルールの情報を元にして正確な判断を下す。与えられる情報があいまいな表現であるときもファジー理論を有効に使って数値に翻訳していく。

知識に基づいて大変正確な判断をくだせる点が長所であり、実用例も多い。例えば金沢観光案内を行うプログラム[7]のデータ選択部分に利用されている。ただし、このAIはルール設定の部分に、扱う分野の専門の知識を必要とするため、手間がかかる上に入力された情報以上の知識を持たないという欠点も併せ持つ。

3．遺伝的アルゴリズム (GA)

この研究では魚のパラメータ調整部分に、複数の数値の組み合わせから結果を得られるという特徴を持つGAを使用することにした。この章では、文献[2][8]を参考に、GAの一般的な仕組みについて述べる。

3.1 基本的な仕組み

1. 遺伝子と個体の生成

各要素が一つながりの情報として数値の配列で表現されたものが、「遺伝子」である。「遺伝子」を持つものを個体と呼び、それを複数個用意する。遺伝子は任意の数を設定しても、乱数によって設定してもよい。数値は2進数がよく使われる。図2-1の各色の長方形が個体である。1、0の数値は遺伝子の各要素の内容を表している。

2. 評価関数による適応度の決定

各個体の評価値を、遺伝子を引数とする評価関数によって算出する。評価値を元に個体の適応度を決定する。

3. 親となる個体と淘汰する個体の選定

新たに作られる遺伝子の元になる個体「親」を選定し、同時に淘汰する個体も選定する。一般的には「親」には適応度が高い個体を選び、淘汰する個体には適応度の低いものを選ぶ。親と淘汰する個体の選択方法には様々な方法がある。各方法については後の章で詳しく述べることにする。

4. 交叉

任意に選ばれた2つの「親」の遺伝子を、途中で交換した個体を作る。この交換作業を「交叉」という。図2 - 2では4つめの要素の後に交叉が行なわれている。作られた遺伝子を持つ個体を「子」と呼ぶ。交叉する場所は乱数によって毎回決められる。何箇所でも交叉するかは場合によって異なる。図2 - 2では1点交叉を行なっている。

5 . 突然変異

「子」の遺伝子に一定の確率で突然変異を起こす。乱数を使って一定の確率で要素を反転させる。突然変異を行わないと進化が止まってしまう。また、突然変異は多過ぎても少な過ぎても良い個体が作られにくいため、その調整は重要である。図2 - 3では赤の位置の遺伝子が突然変異によって反転（1なら0、0なら1になる）している。

6 . 世代交代

「子」を淘汰する個体と入れ替え、新しい世代とする。

7 . 2 ~ 5を任意の回数または目的にあった結果の個体が出るまで繰り返す。

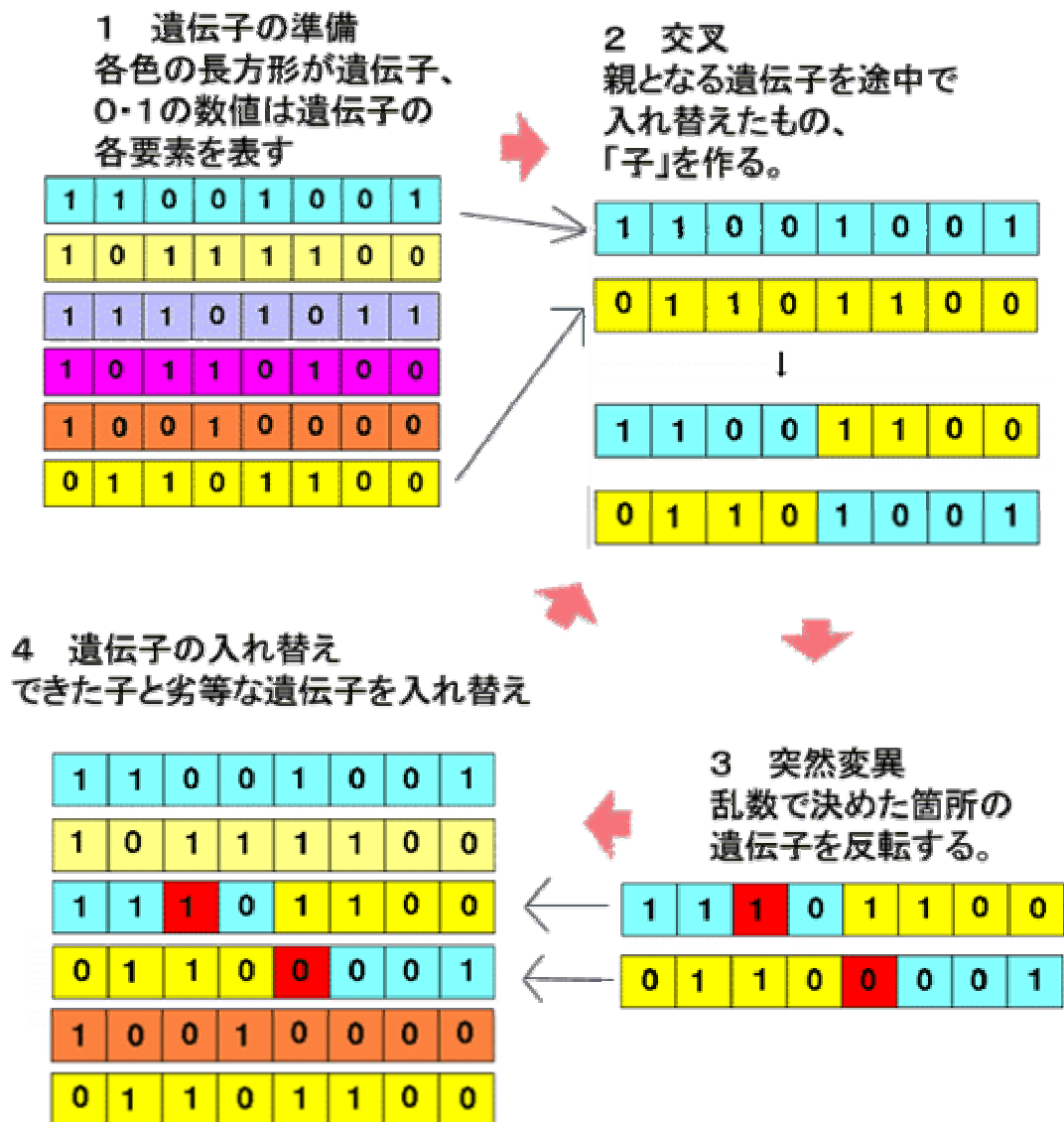


図2 GAのしくみ

赤い矢印が全体的な流れ、黒い矢印が遺伝子の移り変わりを示す。

3.2 世代交代の手法

世代交代の手法については様々な提案がなされている。これらの試みは当初、最適解への収束速度の向上のみを目的としていた。しかし、難易度の高い問題を解かせるにあたり適応度は高いが、一番相応しい最適解とは別の結果になっている「局所解」に陥る事が問題点としてあがってきた。そこで近年では、親の習性を残す「形質遺伝」と、広い範囲での探索をする事で局所解に陥る可能性を少なくする「多様性維持」の2点を重要視したGAの手法の研究が行われるようになってきた[9]。この2点に最も深く関与しているのは親と淘汰される個体の選択であり、様々な方法が提案されている。また、それぞれの選択方法には問題に対する向き、不向きがあり、問題の性質によって使い分ける必要がある。まず主要な選択方法[10]を以下に述べる。

・ルーレット方式

各個体の結果から適応度を出し、選択確率を適応度に比例して設定する。この確率をもとに親となる個体を乱数で選び出す。収束は早いですが、集団の多様性が失われ局所解に陥る「進化的停滞」が起きやすい。

・ランキング方式

各個体を適応度の大きい順に並べる。あらかじめ順位に応じた関数でそれぞれの親から残す子供の数を決めておく。ランクの決定にソートが必要であり、個体数を決める関数、端数の割り当て方法を用意する必要がある。収束速度は速いが、多様性が維持されにくい。

・エリート選択

適応度が最も高い個体を常にコピーして残す。この方式は一般に探索能力が優れているため簡単な問題なら素早く答えを出すのに適している。しかし、親のソートが必要な事、難解な問題になるにつれ探索が局所解に陥り進化に停滞をおこす場合がある事、という問題が残る。

・近親者の相殺

入れ替えの時、できあがった子と似た遺伝子を持つ親と置き換える。多様性の保持に役立つが、収束は遅くなる

- ・トーナメント方式

集団N個の個体群から k 個をランダムに選び出し、その中から最高の個体を一つ選ぶ。この操作をN回繰り返して、選び出された個体だけで個体数N個の新たな集団を作る。この選択方式は計算量の多さや収束性の観点からさまざまな議論がなされている。淘汰される親の選択については次に記す「エリート選択」や「近親者の相殺」などがよく使用され、選択によって多様性維持や収束速度はさまざまである。

- ・ランダム選択

適応度に全く関係なくランダムに選択する。多様性維持はなされるが、収束速度はやや遅い。

ルーレット選択やランダム選択などは適応度の低い個体が残る確率があるが、エリート選択は適応度の高い個体しか残らない。結果的にエリート選択は優勢な遺伝子が全体に広まるのは最も早いですが、残す親の数は多いほど局所探索能力が増す。

GAには、これら選択方法と上書き方法の組み合わせである「世代交代モデル」がいくつか提案されている[11]。代表的なものには次のようなものがある。

- ・ Simple GA (SGA)

ルーレット選択で親を選びランダムに選択した物と子をいれかえる。

- ・ Steady State (SS)

ランキング選択で親を選びエリート選択で淘汰するものを決める。

- ・ CHC

ランダム選択で親を選び、作られた子ともとの親の中から最良2つを残す

- ・ I G S

ランダム選択で親を選び、親集団の適応度の低い個体と子個体を入れかえる。

などである。

最適解を導く速さを重要視するか、多様性の維持を重要視するか、進化的停滞の起きにくさを重要視するか、それぞれの問題によって最適解の出しやすさがあるので、問題の性質や難易度と目的に沿った選択方式を選ぶ必要がある。

3.3 特徴

G Aには様々な特徴があり、それらを生かして様々な問題を解くのに役に立てられている。

- ・ **評価基準を与えるだけで優良な解答を出す**

良し悪しの評価基準だけで良い解答により近づけてゆく事ができるので単純なルールから良し悪しを判断する事象を解く時に便利である。

- ・ **様々な答えの中から最も良い組み合わせを見つけ出す**

一連の組み合わせ情報を一つの遺伝子として格納する為、要素要素に関連性のある数値の生成などに適している。

- ・ **環境などの条件に応じた解答が出せる**

判定の際に環境などの条件を情報として与える事によって、その条件にふさわしい解答を導く事を可能にする。

- ・ **進化の途中段階では良い個体も悪い個体も産まれる。また、最終結果が同一とは限らない**

最初に個体を用意する際や、途中の交叉と突然変異の際に乱数を使用している。その為、途中段階で産まれる個体が必ずしも良い個体とは限らない。また、解く問題に

よっては最終結果が毎回異なる。このようにGAにはランダム性がある。ランダム性は、複数の個体に対して個性有るパラメータの組み合わせを作成するときや、ゲーム上などで強さに故意にばらつきを与えたい時に役立つ。

GAはこのように、判断基準がきまっている事に対してそれに応じた答えの組み合わせを出す事が得意である。

3.4 これまでに成された研究と応用

GAは、上のような特徴を利用し、様々な組み合わせ問題などに役立てられている。

決められた条件や制約の元、最適な組み合わせ、順番などを見つける問題

・「ナップサック問題」

ナップサックに入る荷物の容量を越えない範囲で重要度の最も高い荷物の組み合わせを決める問題[12]。

・「巡回セールスマン問題」

都市とその間の距離の情報から最適な巡回経路を見つけ出す問題[13]。

・「ナーススケジューリング問題」

病院の看護婦の月ごとにおけるスケジュールをいくつかの制約のもとで最適に決定する問題[14][15]。ナーススケジュール問題は「必要日勤者数」「必要深夜勤者数」「必要準夜勤者数」「休みの希望」「指定公休日数」「日勤-深夜勤の組み合わせの禁止」「準夜勤-日勤の組み合わせの禁止」「夜勤パターンの直前の連休は禁止」という実に多くの制約がある。このような制約が多く発生する組み合わせ問題へのGAの使用は、最適解にはかなりの精度が必要であり世代交代数や個体数を多く必要とする。

このようなAI分野でも比較的良く扱われる組み合わせ問題に関しては、GAの分野では主に、それぞれにあった世代交代の手法が研究され続けている。

環境が変化する中でのG Aの適用

- ・「動的巡回セールスマン問題」

巡回セールスマン問題において、途中で都市の位置を変化させた場合への適応をさせる研究[16][17]。

- ・「地形に対する経路制御」

G Aによる経路制御最適化手法の既存の研究[18]。画面上に移動コストの高い場所と低い場所を用意し、30世代ごとに、今まで良い経路が地形の変化でコストの高い経路になるように変更される。この研究では画面を柵目に区切り制御ポイントとなる中継点をいくつか作成し、中継点を結ぶような経路を作成している。中継点の移動や組換えを交叉・突然変異を使用して行う事で最適化している。最適化には成功したが、地形の変化にすぐには対応できない事が結果として指摘されている。

エンターテインメント分野への応用

- ・「G Aによるインタラクティブな敵の進化」

エンターテインメントの分野としては1998年にG Aを使用したゲームが発売されており、手法に関する論文が発表されている[2]。このゲームでは敵一体分のパラメータを一つの個体としている。プレイヤーが毎回しかける数種類の罠を、敵が能力を上げることによって回避できるように成長していく。その敵にさらに対抗できる罠のパターンを考え、再びしかけるといようなゲームである。敵の遺伝子の能力を上げるのにG Aが使用されている。このゲームではG Aの「条件に合わせた進化が出来る事」と「進化の途中段階では良い遺伝子も悪い遺伝子も産まれる事」がうまく利用されている例と言える。

- ・「ロールプレイングゲーム(RPG)制作における敵の強さのチューニング」

実用化はされていないが、上と同じ論文[2]で提案されていた新たなG A適用分野である。RPGゲームなどで配置する敵の強さ(パラメータ)は人の手で行われ1~2ヶ月の期間を要する負荷のかかる作業である。昔に比べて技術の進化に伴いパラメータの数も増えてきており、より大変な作業となってきている。G Aは適切な適応度関数を与える事ができれば、短時間でバランスの良いパラメータを自動的

に生成する事が出来る。

このようにGAは現在でも様々な分野で利用されている。これからはコンピュータの高速化によって、より多くの分野での活用が期待できる。

4 . GAを用いた魚モデルの経路制御

4 . 1 エージェントモデルにおけるGAの適用

ユーザーの望む動きを自動的に行ってくれるモデルを「エージェントモデル」と言う。エージェントモデルの動作には画面からはみ出ない事、障害物と衝突しない事などが必要とされる。また、空間の環境が変わるとパラメータを設定しなおす必要がある。この研究では3DCG空間でエージェントモデルを動作させる際に、エージェントのサイズや移動速度などのパラメータを調整することによって経路制御を行う。これにより障害物等の空間の環境が変わっても障害物に衝突しないエージェントモデルの自動生成を可能にした。今回は一匹の描画にとどめたが、複数個のエージェントモデルを単一のGAによって一度に自動生成することも可能である。

今回の研究をするにあたって3D空間で制御を行なう際に、上下左右方向への移動が可能なエージェントモデルである事から魚モデルへGAを適用した。また、GAはパラメータ数が多くなると極端に計算時間がかかるアルゴリズムである為、パラメータ数は極力少なく抑える必要がある。よって、重力などの極端な外からの力を考慮しない事とした。

4 . 2 魚モデルの経路制御パラメータ

魚の行動は「通常の直線的な動き」と障害物が視界に入った時の「障害物回避の方向転換と速度変化」という2つの行動パターンをあらかじめ設定しておく。「2つの行動パターンとその切り替え部分の数値の情報」のみで経路に変化を与えるようにする事で、経路制御をGAで扱うことを可能にした。障害物は水槽の端の壁、孤立物体（球、立方体）の3種類とした。

遺伝子は水槽の広さや障害情報などの環境に適応する「回避に使用する情報」と魚自体のサイズや色といった「モデルの個性」を一つながりの情報として格納している。具体的

な遺伝子の内容は、格納されている順に、

- ・ 魚の初期位置 (x,y,z)
- ・ 魚モデルの拡大率
- ・ 魚から見える視界の距離 (視界の中に障害物があると回避行動をとる)
- ・ 回避行動時に回転する角度
- ・ 最高速度 (標準時泳がせる速度)
- ・ 魚の初期移動方向
 - ヘディング角 (水平方向の角度)
 - ピッチ角 (垂直方向の角度)
- ・ 色 (4種類の中から選ぶ)
- ・ 回避行動時に良く回転する方向 (右または左の2種類)

となっている。

*なお、魚モデルの拡大率とは、魚モデルの大きさを変える為に用意したパラメータである。描画する時に同一の3D魚モデルを使用するため、大きさを変えるには元となる魚モデルの「縦」「横」「高さ」の比に拡大率を掛けて、各辺を同じ比率で拡大する。

障害のある環境で経路制御するにあたっては、魚のサイズと障害物を判定するための視界、初期位置の座標などはそれぞれ関連性が高いパラメータである。水槽内の情報(障害物の位置や大きさ、壁の位置)とも関連性が高い。したがってGAを使用して調整を自動で行うに相応しい部分であると言える。

1と0以外の数値を必要とするパラメータに関しては8ビットで表現し、2進数から10進数に変換する事で幅を持たせた。ただし、2進数から10進数への単純な変換では左右の桁の重要度が極端に違ってしまいGAで扱うには相応しくない。そこで各桁の表現を任意の整数値を加えるか否かとする事で解消した。数値はそれぞれ1、5、17、26、24、16、9、2とし、これを合計すると100になり、数値の組み合わせで0~100までの自然数をほぼ全て表現できる[3]。この自然数に対し各パラメータごとに一次関数で計算し、それぞれ適性範囲に収まるようにしている。範囲を限定することで無駄な遺伝子が出る可能性を低くし、GA部分の世代交代を減らした結果として高速化にもつなが

った。

各パラメータの取る値とビット数は次の表 1 の通りである。

表 1 各パラメータの値範囲と占有ビット数

パラメータと単位	範囲	ビット数
初期座標(x,y,z) (ピクセル)	水槽の端から 40 ピクセルより内側となる座標値の実数	各 8 ずつ
魚モデルの拡大率 (倍)	2 ~ 17 の実数	8
視界距離 (ピクセル)	20 ~ 100 の実数	8
回避行動時の回転角度 (度)	45 ~ 215 の実数	8
最高速度	5 ~ 45 の実数	8
初期方向 (ラジアン)		
ヘディング角 (水平方向)	0 ~ 2 の実数	8
ピッチ角 (垂直方向)	-1/6 ~ +1/6 の実数	8
色	0 ~ 4 の自然数	2
回避時によく回転する方向	0 と 1 の自然数	1

これら計 75 ビットの 2 進数の数列で一匹分の遺伝子が表現されている。回避行動については 4.5 節で詳しく述べる。

4.3 GA による経路制御パラメータの最適化

GA の遺伝子生成数は必要となる計算量を考慮し 10 個体分とした。遺伝子の優劣を判定する評価関数は魚と障害物の衝突回数を採用し、少ないほど優秀な個体とした。衝突判定に関しては衝突判定の章で詳しく述べる。

世代交代のモデルは、最適解を素早く得る事が出来る「Steady State モデル (SS)」を参考にしている。最も成績の良い遺伝子を親として選択し、最も成績の悪い遺伝子を淘汰する。このモデルには局所解に陥りやすいという弱点があるが、試行の結果、最適解を得られる世代交代数が、多くても 2 桁で収まる事を確認した。

交叉と突然変異は、「パラメータフリー GA (Pf-GA) [19]」を参考に、ランダムな位置での 1 点又は 2 点交叉と、全ての遺伝子に対して一律 0.06 の確率で突然変異を起こさせる事にした。

Pf-GA とは交叉の位置を完全にランダムにし、どのビットにも一律に突然変異を起こさせる GA である。様々なところに変化を起こさせる事で最適解の局所的な探索も全体的な探索も行う為に探索能力が優れている GA である。また、パラメータごとの適応度

の判定や突然変異率を指定する必要がないので、今回のような各要素の総合的な結果から評価を行うようなGAには向いている方法である。

4.4 衝突判定

衝突判定は次のように行った。まず、魚の中心と上下左右前後の端に図3で示すようにポイントを置く。各ポイントが以下の状態の場合に衝突したとみなすこととした。

- ・ 孤立物体と重なる。
- ・ 水槽の端と重なる。
- ・ 水槽の端より外に出る。

* ここで孤立物体とは水槽中での岩や障害物の事を表す。

衝突回数は各ポイントごとに計算し合計数を出す。

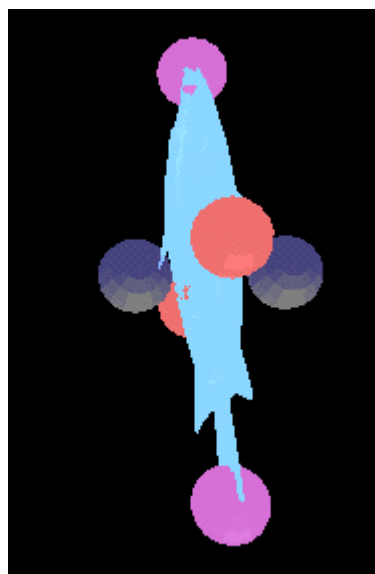


図3 - 1 魚と衝突判定用ポイント 図3 - 2 下から見た、魚とポイント
赤、水色、紫の球が衝突判定用ポイントになっている。

衝突判定はそれぞれ以下のように実現されている。

- ・ 孤立物体

立方体 - 魚のポイント座標が物体の内側に入ると衝突していると判断する。

球 - 魚のポインタから球の中心の距離が半径より短いと衝突していると判断する。

・ 壁

魚のポインタ座標が壁ある X、Y、Z の各座標よりいずれかが外側にあれば衝突していると判断する。

4.5 視界による障害物判定と回避行動

GA の遺伝子では数値情報しか扱えない。GA で経路を調整するには、数値だけで個体の動きに個性を与えられる方法で行動の制御をする必要がある。しかし、先にも述べたように GA は扱うパラメータ数が多くなると極端に計算時間が増大する。そこで通常行動・回避行動・行動の切り替えをパターン化しておき、要所要所に遺伝子からパラメータを与えるだけで回避軌道に変化を与えるようにしておく。この方法によってより少ないパラメータによる経路制御を可能にした。魚の回避行動は次の通りである。

1. 視界による障害物の検出

魚の中心から各障害までの距離が視界距離より短い時、視界に入ったとみなす。

なお、立方体に関しては「物体の中心から端までの距離」を辺の長さの $\sqrt{2}/2$ とする事で球と同じように扱っている。視界判定は孤立物体の中心が、魚が向いている方向から上下に 45 度、左右に 45 度にあるものを判定対象にする。

前後左右の壁は、向いている方向から左右に 45 度以内に存在するもの 1 方向または 2 方向を判定対象にする。上下の壁は魚の向きが少しでも上下に傾いていれば判定対象にする。

2. 回避行動

回避行動は障害物に応じて次に示す 3 パターンに分かれる。

a. 障害が孤立物体または水平方向の壁一方向のみの場合

1 フレームにつき直進する速度を 0.01 落とし (ただし最低速度を 5.0 とする) ヘディング角を 10 度回転する。これをパラメータで定めた回避角度分までつづける。魚モデルは図 4 で示すような軌道を得て壁を回避する。図 7 は実際に奥の壁に対して回

避行動を取っている魚の正面からのスクリーンショット、図8は魚モデルが右の壁に対してとった回避軌道を上から見た図である。

b. 障害が水平方向の壁（2方向以上）の場合

1フレームにつき直進する速度を0.01ずつ落とし（ただし最低速度を5.0とする）ヘディング角を10度ずつ回転する。これをパラメータで定めた回避角度分+30度までつづける。魚モデルは図5で示すような軌道を得て壁を回避する。

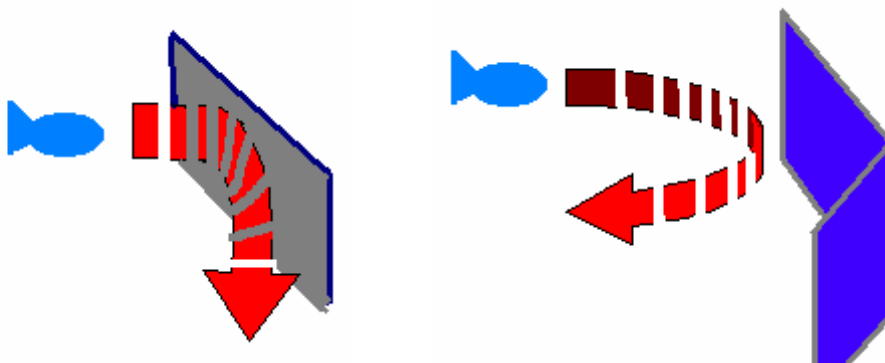


図4 孤立物体又は水平方向の壁に対する回避行動 図5 二方向の壁に対する回避行動

c. 障害が上または下の壁のみの場合

速度は変化させない。壁のある方向にピッチ角（魚モデルの上下方向の角度）が向いていた場合のみ、ピッチ角を1フレームにつき5度分ずつ、9フレーム（45度）反対方向に回転する。魚モデルは図6に示すような経路を取って壁を回避する。

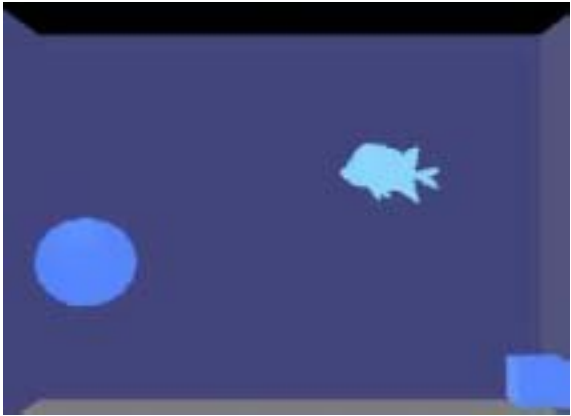


図6 上下の壁の回避

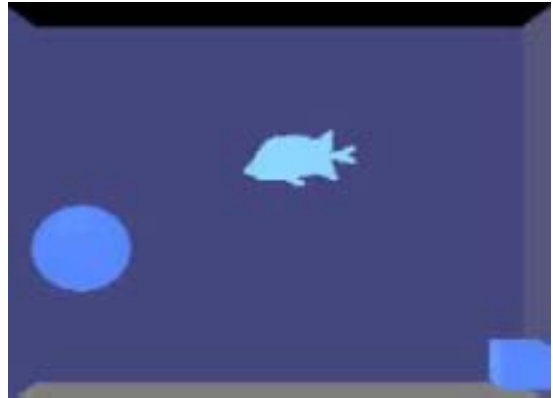
* 孤立物体や水平方向の壁に対しての回避で回転する方向は、遺伝子に指定しており、回転の確率が指定方向：反対方向で3：1になっている。速度を上げていき最高速度に戻る。

3 . 速度を戻しつつ障害が再び視界に入るまで直進を続ける。

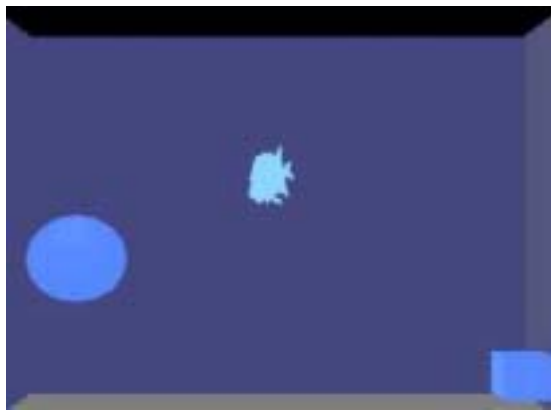
最高速度に戻るまで速度を1フレームにつき0.1ずつ増加させる。さらに、一定コースにとどまる事を防ぐ為、直進50回につき一回、描画回数に応じて上下の方向にわずかに傾く「揺らぎ」を与える。



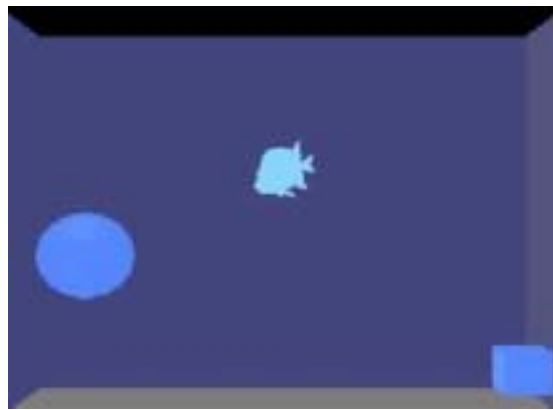
7 - 1



7 - 2



7 - 3



7 - 4

図7 奥の壁に対して回避行動を取る魚



図8 右の壁に対して回避行動を取る魚と通った軌道（上から見た図）

4.6 魚モデルの遺伝子の設定と画面描画

ここでは遺伝子作成から描画までの全体の流れを述べる。

まず、魚モデルが泳ぐ環境を設定する。水槽の大きさを決めておき、端に壁を配置する。障害物である孤立物体の形（球か立方体か）・個数・位置・サイズを設定する。

次に乱数で個体の遺伝子を複数個設定する。この遺伝子の2進数の情報から魚モデルのそれぞれのパラメータ（初期位置、スピード、大きさ、回避行動など）で扱う範囲になるよう変換してから設定する。

壁の場所や1回の描画で進む速さ等を、実際の描画と同じ状態に設定して魚モデルが辿る経路を計算し、衝突判定を行なう。ここで算出される衝突回数を遺伝子の評価関数とする。この研究ではパラメータの設定のみで自動的に動き続ける事を想定しているので、規定の描画回数は存在しないが、評価関数としての信頼性を考えると、より多くのフレームを計算し衝突判定を行なう事が望ましい。そこで、必要となる計算量を考慮し、判定用に計算するフレーム数は20000とした。

泳ぐ経路は基本的には直進で、1フレームにつき0.25×速度分移動する。視界内に障害物が入ると回転して回避する。この設定のみで最適化を試行したところ一箇所を回転し続ける個体が出現した。そこで、泳ぎに一定の確率で「揺らぎ」（魚の直進の描画50回に1回、描画回数に応じて上下のどちらかへ向きを少し傾ける）を与える事で、ある程度解消する事にした。

成績を元に遺伝子の選定を行ない、「交叉」「突然変異」「入れ替え」を行なう。衝突回数が0回になる遺伝子が産まれるまで、遺伝子に入れ替えから選定までを繰り返す。

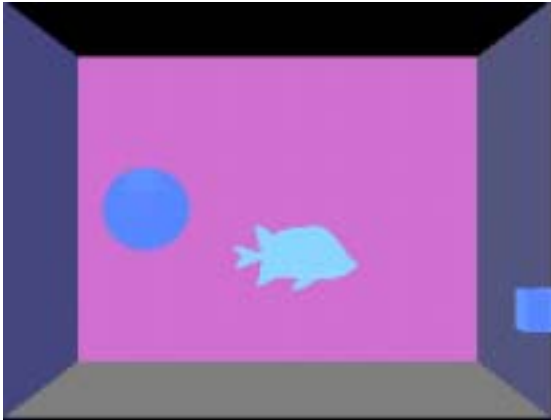
5. 結果

GAによってパラメータの調整をした結果、衝突判定で遺伝子の良し悪しを判断するだけで、障害に衝突しない「魚モデルの経路制御」が行われた。

同一環境下で20回描画を行い、毎回異なるパラメータが得られることを確認した。得られたパラメータを表2に示す。同一環境下で作成されたパラメータから実際に描画したものが図9である。

表 2 横 300 縦 400 奥行き 300 の水槽環境で作られたパラメータの組み合わせ

サイズ	視界距離	最高速度	角度	GA世代交代数
6.86	56	11.8	175	18
5.42	85	16.2	126	24
4.52	78	11.4	124	4
5.96	75	13.4	209	8
9.74	90	14.2	201	1
8.12	52	15	179	9
9.92	76	12.6	118	21
5.78	54	27	177	7
4.7	76	12.6	158	28
6.86	77	23.4	133	10
5.96	86	11.8	136	7
6.32	81	24.2	140	9
7.76	73	16.6	113	7
11.36	90	7	174	0
5.6	67	12.2	168	14
8.3	59	27	116	6
6.14	96	19.8	133	4
6.86	65	14.6	172	2
6.32	69	17.8	89	12
2.18	48	12	82	5
6.734	72.65	16.05	146.15	9.8



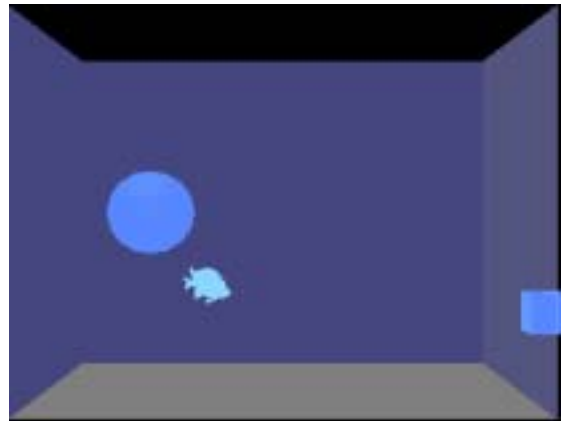
9 - 1



9 - 2



9 - 3



9 - 4

図9 同じ環境で出来た魚モデル 同じ環境でも大きさ・色などが様々である。

次に、「次に、環境の変化に応じた最適化が行われていることを確認するために、水槽の広さを3種類用意し(孤立物体の位置は同じ)、それぞれ20回ずつ試行した。それぞれの環境に合ったパラメータが生み出された。その際、水槽を狭い空間にした場合では表3が示すように遺伝子の世代交代数の平均が増え、広い空間にした場合は減った。また、表4、5が示すように各パラメータにも画面の大きさに関係して一部の平均値が変化するような傾向が見られた。表4の網掛け部分には水槽の大きさの変化に対し、単調増加、速度は単調減少の傾向が見られた。

画面サイズはそれぞれ

小 300 × 400 × 300

中 400 × 600 × 300

大 800×800×800 である。

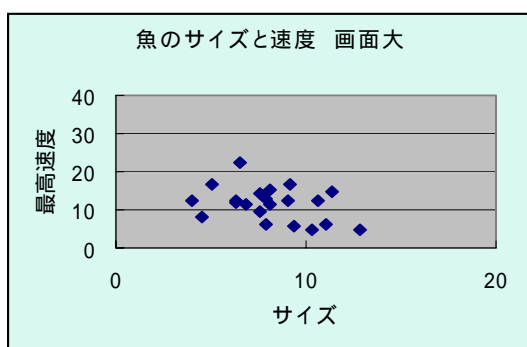
表3 水槽の広さを変えて20回パラメータを作成した時の各世代交代数の内訳

世代交代数 (世代)	水槽 小	中	大
0～5	6回	13回	12回
6～10	8回	2回	4回
11～15	2回	1回	4回
16～20	1回	0回	0回
21～	3回	4回	0回

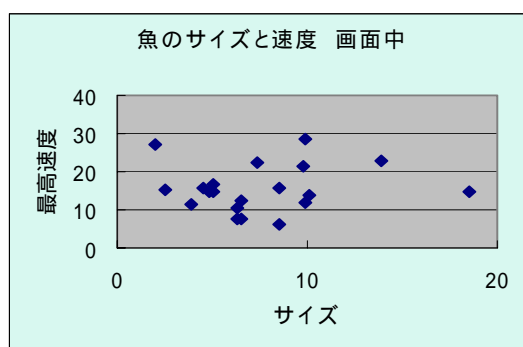
表4 パラメータ平均値

	小	中	大
サイズ	6.734	7.503	8.12
視界	72.65	78	78.05
速度	16.05	15.62	11.58
角度	146.15	151.9	147.3
世代	9.8	7.4	5.25

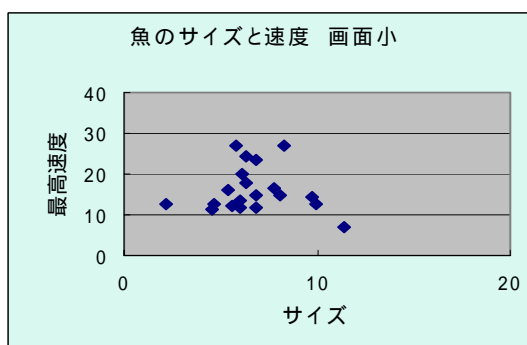
表5 画面（水槽環境）の広さ別、魚のサイズと速度



5 - 1



5 - 2



5 - 3

表5からは画面のサイズによって、ばらつきの中心位置がわずかに違っている事がわかる。

これらの平均値の違いなどの結果から、環境の広さに合わせたパラメータ設定がなされていると言える。また、狭い水槽ほど最適化に必要な世代交代数が増えたのは、狭い環境では壁に当たりやすく、より細かいパラメータの調整が必要になった為と思われる。

6. 考察

GAによって衝突しない経路制御という目的は達成する事が出来た。しかし、視覚的な面ではまだまだ課題が残っている。今回の研究では制御パラメータ数の削減のため、魚モデルの回避行動を単純化した。その為、回転角度が急過ぎたり減速加速が不自然だったり、視覚的に不自然な点がでてしまった。また、行動に揺らぎを与えてあっても、魚のサイズが巨大な場合や視界距離が広すぎる場合においては

- ・ 同じ経路を何度も辿り続け一定の位置しか移動しない。
- ・ 常に回避行動をとり続ける為に、円を描きつづける。

といった現象が起きた。この現象の改善方法として揺らぎを大きくする事が考えられる。しかし、揺らぎを大きくして実験した結果、今までは衝突しないような場面で揺らぎによって衝突が起こってしまった。

通常ではGAによって正しいパラメータ設定が行なわれるが、無理な設定をすると正しく衝突判定が行われない事や、パラメータの設定がうまく行なわれない事もあった。無理な設定とは主に次のような場合である。

- ・ 大きすぎる障害物の配置や狭すぎる水槽環境により適応パラメータが存在しない。
- ・ 広すぎる水槽で泳がせた場合の経路計算回数の不足。
- ・ 魚のサイズの極端な巨大化による衝突判定箇所の不足。

ランダム性に関しては同じパラメータを持つ個体がなかったことから、多数の個体を必要とする場合に個性ある個体を作る事に役立つ。この性質は、動物の群れなどの、個体同士で似てはいるがそれぞれ少しずつ違う動きをするというような集団モデルのパラメータ

調整に等にも大いに役立つ。

視覚的な面に残された課題については、パラメータ数を増やして回避行動のパターンをより細かく設定する事や、各パラメータの取る数値幅を狭く調整することによってある程度改善されると思われる。また、実際の魚がとる行動パターン「壁や物体に沿って泳ぐ」「水面の直下を泳ぐ」「底をつついて餌を探す」「一箇所にとどまる」などは今回は実装しなかった。今後、よりリアルさを追求するのであれば回避行動以外にもこのような魚によく見られる行動パターンのアルゴリズムを考え、魚の種類によって選択して増やす必要があると考えられる。

7 . 謝辞

本研究を進めるにあたり、ご指導いただきました東京工科大学メディア学部の渡辺大地先生、副査を担当して頂きました和田篤先生に深く感謝いたします。また、本研究で使った魚モデルは東京工科大学アクアプロジェクトより提供されています。快く協力して頂き感謝いたします。

8 . リファレンス

[1] 鳩宿潤二,中嶋正之,高橋裕樹：“動画像を用いたバーチャルフィッシュの動きの生成”

第 14 回 NICOGRAPH / MULTIMEDIA 論文集 p73,1998.

[2]森川幸人：“テレビゲームへの人工知能技術の利用”

人工知能学会誌, vol.14, no.2, p214, 1999.

[3]森川幸人：“マッチ箱の脳”,新紀元, 2000.

[4]伊庭斉志：“遺伝的プログラミングと進化論的計算手法”,

人工知能学会誌, vol.19, no.4, p512, 1994.

[5]佐々木 崇, 伊庭 斉志, 石塚 満：“遺伝的プログラミングを用いた実時系列データ予測”

第 56 回情報処理学会全国大会予稿集(2), p.189, 1999.

[6] 塚本 壮輔, 大本 祐栄, 酒井 正直, 原 肇：“ニューラルネットワークによる音声認識”

広島大学工学部, 電子工学科, 卒業論文, 1996.

- [7]岸本英明:金沢市観光計画支援システム「まわるまっし金沢」,
まわるまっし金沢 NetWork, <http://www.fuji.ne.jp/~kissy/kanazawa/index.html>, 1996.
- [8]伊庭斉志:“遺伝的アルゴリズムの基礎”,オーム社,1994.
- [9]山村雅幸, 小林重信:“遺伝的アルゴリズムの工学的応用”,
人工知能学会誌, Vol.9, No.4, p506, 1994.
- [10]北村新三:“遺伝的アルゴリズムについて”,
<http://www.al.cs.kobe-u.ac.jp/~omori/study/GAsec1.html>.
- [11]佐藤浩, 小野功, 小林重信:“遺伝的アルゴリズムにおける世代交代モデルの提案
と評価”, 人工知能学会誌, Vol.12, No5, p.734, 1997.
- [12]有光郷司:“ナップサック問題への遺伝的アルゴリズムの適用”
ARIMITSU SATOSHI'S WebSite, 研究用資料, 1995.
<http://www.st.rim.or.jp/~arimitsu/Japanese/oldpages/study/ga-1.html>
- [13]秋場耕一:“遺伝的アルゴリズム”, 山形大学, 教育学部, 卒業研究, 1994.
- [14]鶴岡信治:“暮らしに役立つ情報処理技術”,
<http://www.ip.elec.mie-u.ac.jp/ja/ppt/rsp/rsp.files/frame.htm>.
- [15]熊谷幸介:“ナーススケジューリング問題への遺伝的アルゴリズムの適用”,
山梨大学, コンピュータ・メディア工学科, 卒業論文, 2001.
- [16] 中谷直司, 吉田等明, 三浦守:“遺伝的アルゴリズムの動的問題に対する適用”,
計測自動制御学会東北支部, 第 194 研究集会資料, p194, 2001.
- [17] 吉田 等明, 和賀 光悦, 恒川 佳隆, 三浦 守:“GAを用いた動的巡回セールスマン
問題へのアプローチ”, 計測自動制御学会東北支部, 第 167 回研究集会資料 167-1, 1997.
- [18]川上孝義:“変化する地形に対する遺伝的アルゴリズムを用いた経路最適化”,
広島市立大学, 知能情報システム工学科, 知能システム講座, 卒業論文, 2001.
- [19] 池田亜未: “パラメータ設定不用の遺伝的アルゴリズム”,
山形大学, 教育学部, 情報教育コース, 卒業論文, 2002.