

Title	GAを用いたプレイヤーのレベルに適應する 多様なオセロAIの開発
Author(s)	上田, 陽平
Citation	
Issue Date	2012-03
Type	Thesis or Dissertation
Text version	author
URL	<a href="http://hdl.handle.net/10119/10441">http://hdl.handle.net/10119/10441</a>
Rights	
Description	Supervisor:池田心, 情報科学研究科, 修士

修 士 論 文

GA を用いたプレイヤーのレベルに適応する  
多様なオセロ AI の開発

北陸先端科学技術大学院大学  
情報科学研究科情報科学専攻

上田陽平

2012 年 3 月

## 修士論文

# GAを用いたプレイヤーのレベルに適応する 多様なオセロAIの開発

指導教官 池田心 准教授

審査委員主査 池田心 准教授  
審査委員 飯田弘之 教授  
審査委員 白井清昭 准教授

北陸先端科学技術大学院大学  
情報科学研究科情報科学専攻

1010008 上田陽平

提出年月: 2012年2月

## 概要

近年は計算機の性能や探索・学習アルゴリズムの発展と向上によって多くのゲームで人間プレイヤーよりも強い人工知能 (AI) を作る事が可能になった。学術的な研究において、特に“強さ”に関する分野は十分な域に達している。

一方、アマチュアが AI と対戦してゲームを楽しむためにはいくつかの課題が残っている。通常のゲームソフトではプレイヤーのゲームの技量に応じて任意で難易度を設定し、何らかの手加減を行うが、それらの多くは静的に設定されたパラメータに基づいているため、プレイヤーに合った強さを持つ相手と戦えることは少ない。更に手加減を行うことによって不自然な手抜きをされることがあり、不快に感じることも課題である。

また、更に強さの異なる AI を用意しても共通のアルゴリズム、設定されたパラメータに基づいて動作しているため、その戦略、弱点が似てしまうという課題もある。人間のプレイヤーからすれば同じ相手と対戦するという事になり、単調で多様性が無く、張り合いが無くなることでプレイヤーのゲームの継続の弊害や、その AI の弱点にのみ特化した戦法を身につけてしまうという問題がある。

これらの問題を踏まえて、ゲーム AI と人間の良い関係が築ける要素として 1. 力量が同程度であること、2. 対等な相手と感じられること、3. 近い力量を持でありながら、違う特徴・戦略を持つことが挙げられ、それらの要素をシステムに実装するために 1. プレイヤーの強さの計測、模倣、2. 不自然な着手選択を抑制する入出力モデル、3. プレイヤーのレベルに対応させるアルゴリズムの技術を考案し、用いた。

システム全体のアプローチは大きく分けて 2 つの段階に分けられ、1. プレイヤーの棋譜をテスター AI が評価し、同程度の力量のエージェント AI を構成し、2. 複数の AI をエージェント AI との対戦結果や戦略の多様性から評価し、遺伝的アルゴリズムを用いて最適化する。これによって、同じ強さを持ちながら GA によって戦略、弱点が違う相手を複数生成できると期待できる。

第 1 段階ではテスター AI と呼ばれる人間よりも強い AI が棋譜を評価し、テスター AI の最善手との評価値の差に基づいて、着手の強さの制限、安定性を維持したエージェント AI を作成した。様々な強さ、手法を用いた複数の AI を対象としてエージェント AI の性能を試した結果、元となった AI と 5 分の戦いが行うことが出来たので強さの模倣には成功した。

第 2 段階ではヒューリスティックな評価によって自然さを考慮したモデルと 3 つの特徴量パラメータから構成された GA の個体群を最適化、エージェント AI との対戦結果が 5 分になるように、特徴量パラメータの異なりが大きくなるようにした。線形和パラメータとよばれる適応度を抑制する重みを調節することで勝率が 0.5、特徴量パラメータが離れた個体を複数作ることに成功し、人間 (中級者) による評価でもそれらの着手、戦略の違いが確認できた。

被験者実験の結果は強さ，着手の不自然は多少の偏りはあったものの本研究としての目的は達成はできた．しかし，特徴，戦略の多様性を感じさせる部分に関しては若干課題が残る結果となった．これは今回用いた特徴量パラメータ（開放度，着手可能点，確定石）が初級者にはやや高度で判断しづらいことによる．そこで，初級者にもより分かりやすい表現を特徴量パラメータとして追加することでこの問題を解消することが今後の課題である．この様に多くのパラメータを追加しても相手の強さに合わせることが出来るのが遺伝的アルゴリズムを用いたこのシステムの利点でもある．

# 目次

第1章	はじめに	1
1.1	背景	1
1.2	目的	2
第2章	ゲーム AI のあり方	3
2.1	ゲーム AI とプレイヤーの関係	3
2.2	ライバル的存在としてのゲーム AI	4
第3章	研究のアプローチ	5
3.1	提案システムの全体像	5
3.2	同レベルのエージェント AI の構成	5
3.3	多様な戦略を持った AI 群の構成	6
第4章	オセロ	7
4.1	オセロの概要と、実験に用いる理由	7
4.2	オセロの状態評価関数と特徴量	8
4.2.1	状態評価関数	8
4.2.2	着手可能点 (mobility)	8
4.2.3	開放度 (frontier)	8
4.2.4	確定石 (stable piece)	10
4.3	オセロにおける不自然な手	10
第5章	強さの計測と同じレベルのエージェント AI の作成	12
5.1	目標とアプローチ	12
5.2	複数の AI の強さの検証	13
5.2.1	方法	13
5.2.2	実験結果と考察	14
5.3	既存方法を用いた AI の検証	14
5.3.1	方法	14
5.3.2	実験結果と考察	16
5.4	提案手法を用いた AI の検証	18
5.4.1	方法	18

5.4.2	実験結果と考察	19
5.5	まとめと今後の課題	19
<b>第6章</b>	<b>多様な AI を作成するアルゴリズムの実装, 検証</b>	<b>22</b>
6.1	目標とアプローチ	22
6.2	遺伝的アルゴリズム (MGG)	22
6.3	BLX- 交叉	23
6.4	実験	24
6.4.1	方法	24
6.4.2	特徴量パラメータと不一致率の関係	26
6.4.3	適応度の推移	26
6.4.4	各個体の 3 次元空間上の分布	29
6.4.5	勝率 0.5 付近の個体同士の着手の相違	29
6.4.6	ブラインドテストによる多様性の評価	31
6.5	まとめ	32
<b>第7章</b>	<b>被験者実験に基づいたライバル AI の検証</b>	<b>34</b>
7.1	目標とアプローチ	34
7.2	実験	36
7.2.1	アンケート結果	36
7.3	まとめ	38
<b>第8章</b>	<b>おわりに</b>	<b>40</b>

# 第1章 はじめに

## 1.1 背景

ゲーム情報学は人工知能分野の1つとして、年々大きな進歩、進化をしてきた。その起源となるのはシャノン [1] が1950年にminmax探索によって局面評価関数値を求め、その値が最も高い手をチェス盤上に打つというものである。これを機にゲーム情報学は多くのゲームにおいて様々な手法が試され、AIは改善改良された。

その1つの例が強さを求める研究であり、その強さはプロ相手でも太刀打ちできないゲームもある。1997年ではチェスではDeepBlueがオセロではLOGISTELLOが世界チャンピオンのカスパロフ氏に勝利を収め、その後も更に改善されて現在では人間が勝つことは不可能となっている。また、将棋は2007年にBonanzaが渡辺明竜王と対局し敗北はしたがタイトル保持者が苦戦したことから将棋AIはプロの域に迫り、2010年は合議制AIのあから2010が清水市代女流王将に勝利、2012年1月にはボンクラーズが元名人の米長邦雄永世棋聖に勝利した。囲碁も2006年にCrazyStoneが出て以来、モンテカルロ木探索が主流となり今は19路ではアマチュア2,3段レベルに向上し、発展が期待されている。

これからの研究の焦点となるのは強さを自動調節するための研究、多様な戦略を求めた研究などが挙げられる。強さを自動調節するための研究とはAIがアマチュアに対してレベルを下げて対戦することに主眼を置いている。現在多くのゲームAIがアマチュアよりも強くチューニングすることが可能であるため、そこからプレイヤーに合わせて対戦することが可能である。2007年にはJongyeolら [2] がオンライン学習によるゲームレベルの動的生成を行った。これはGMMと呼ばれるモデルによってユーザのプレイパターンの確率を分析し、その後生成されたスクリプトを用いて敵の強さの重みを動的に変えていくというものであり、2Dシューティングゲームを用いて実装されている。また、加藤ら [3] はTD( )法を用いて将棋で人間の手に合わせた局面評価関数の微調整を行った。

一方、多様な戦略を求めた研究というものはAIそれぞれに多様な戦略、個性を持たせることにより幅の広いプレイングを実現するためのものである。一般のゲームではAIはそれぞれ決められたパラメータに基づいて行動が決定するが、これはゲームAIの戦略の固定化につながり、プレイを続けるうちに何度も同じ相手と対戦しているだけでプレイヤーのスキルが上がらない、モチベーションが続かないなどの点からこの研究は必要とされている。中川ら [4] はニューラルネットワークを用いて格闘ゲームAIの難易度調整を行いつつ、様々なアクションを実現できるように改良、多様性の向上につながった。また、Pieterら [5] はDynamic Scriptingと呼ばれる動作を記述したルールベースをそれぞれの



キャラクターに割り当て、行動の多様性の実現に図った。

本研究で扱うオセロにおける人間のレベルに適応させるため、自然な動きを行うための関連研究は少なく、GA[6][7][8] やニューラルネットワーク [9] などによってより強い戦略を作り出す研究がほとんどである。また、遺伝的アルゴリズム (GA) を用いたフリーのオセロソフト”Darwersi” [10] は先生のようにプレイヤーを指導するために作られている。このソフトウェアはダーウィンの進化論を基にしており、ゲーム開始時の展開や終盤時の詰めの教育、着手の分析を行う。

## 1.2 目的

本研究では、プレイヤー（初級者）の”ライバル”となる AI を生成することを目的としている。ライバル（好敵手）とは“ 勝負事などで力量がつりあったよい相手（広辞苑より）”と定義され、それに基づき以下の3つのライバルの特徴を提唱し、これを重点においたシステムを提案する。

- 力量が同程度であること
- 対等な相手と感じられること
- 近い力量を持でありながら、違う特徴・戦略を持つこと

## 第2章 ゲームAIのあり方

### 2.1 ゲームAIとプレイヤーの関係

近年は計算機の性能や探索・学習アルゴリズムの発展と向上によって多くのゲームで人間プレイヤーよりも強い人工知能 (AI) を作る事が可能になった。1997年にはチェスで Deep Blue が世界チャンピオンのカスパロフ氏に勝利し、最近では将棋においてあから 2010 が清水市代女流王将に、ボンクラーズが元名人の米長邦雄氏に勝利した。この様に学術的な研究において、“強さ”に関する分野は十分な域に達している。

一方、アマチュアが AI と対戦してゲームを楽しむためにはいくつかの課題が残っている。競技用が開発された AI は高度な戦略を展開し、一般の購入者では勝つことができない。そのため、通常のゲームソフトではプレイヤーのゲームの技量に応じて任意で難易度を設定し、何らかの手加減を行う。手加減の方法は探索の深さを浅くする、評価関数に乱数を加えるなど様々であるが、それらの多くは静的に設定されたパラメータに基づいているため、プレイヤーに合った強さを持つ相手と戦えることは少ない。更に手加減を行うことによってプレイヤーにとって不自然な手抜きをされることがあり、不快に感じることも課題である。

更に強さの異なる AI を用意しても共通のアルゴリズム、設定されたパラメータに基づいて動作しているため、その戦略、弱点が似てしまうという課題もある。人間のプレイヤーからすれば同じ相手と対戦するという事になり、単調で多様性が無く、張り合いがなくなることでプレイヤーのゲームの継続の弊害や、その AI の弱点にのみ特化した戦法を身につけてしまうという問題がある。また、個性のある AI が複数用意されている場合もあるがそれを手作業で調整することは高コストである。これにより、特徴、戦略の自動化が求められ、パラメータの調整ができれば攻めに特化、守りに特化した AI など多様な相手との対戦が行えるので、相手の手筋に対して臨機応変な対応が身に付くなどの利点が得られる。

これからのゲーム AI と人間の良い関係が築ける要素として以下のものが挙げられ、本研究では主に多様な特徴と強さの調整に着目している。

- 多様な特徴  
異なる特徴を持った相手と対戦できること
- 強さの調整  
自分と似た強さの相手が自動生成されて戦えること

- 自然な動き  
着手が手加減しているものであれ違和感を覚えさせないこと
- 教育的な指導  
技術や戦略を上達するための指導が行えること

## 2.2 ライバル的存在としてのゲーム AI

2.1 節を踏まえて、目指すべきライバル AI の特徴として以下のものが挙げられる。

- 力量が同程度であること
- 対等な相手と感じられること
- 近い力量を持でありながら、違う特徴・戦略を持つこと

この様な AI もしくは AI 群を構成するために、本研究では以下の 3 つの要素技術からシステムを構築する。

- (a) プレイヤの強さの計測，模倣をする技術  
十分に強い AI の最善手とプレイヤの着手の評価値の差を計測，強さを模倣を行う
- (b) 不自然な着手の抑制をする技術  
初級者が不自然と感じる手と理由を分析し，それをフィルタする入出力モデルの構築を行う
- (c) プレイヤと同じ強さで，かつ多様な戦略を持つ AI 群を構成する技術  
2 種類の評価値に基づく遺伝的アルゴリズムを用いた AI 提示システムの構築を行う

このシステムを用いて，プレイヤに自分と同じレベルの AI と対戦できるだけでなく，多様な特徴を持つ複数の AI と対戦できる環境を与え，モチベーションを維持して技量の向上を図る。

本システムは様々なゲームに適用できるが，今回は既に多くの人間よりも強い AI が公開されており，競技人口が多くルールが単純なオセロを対象として実装と評価を行う。

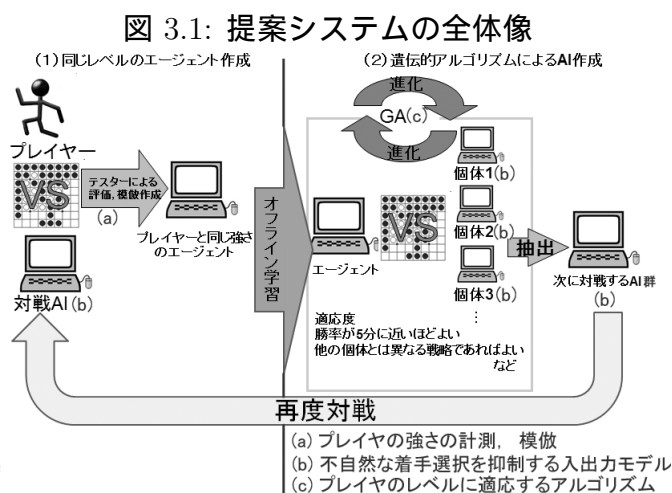
# 第3章 研究のアプローチ

## 3.1 提案システムの全体像

本研究では、人間プレイヤーと同じの強さを持ちつつ、特徴が異なり、不自然な行動をしないAIを複数構成するシステムを提案する。提案システムの全体像は図3.1に示すが、これは大きく分けて以下の2つの段階から成り立っている。

1. プレイヤの棋譜をテスター AI が評価し、同程度の力量のエージェント AI を構成する
2. 複数の AI をエージェント AI との対戦結果や戦略の多様性から評価し、遺伝的アルゴリズムを用いて最適化する

システムを2段階に分けることにより同じ強さを持ちながら GA によって戦略、弱点が異なる相手を複数生成できると期待できる。



## 3.2 同レベルのエージェント AI の構成

段階 (1) では (a) プレイヤの強さの計測模倣の要素技術に基づき、プレイヤーと同じレベルのエージェント AI を構成する。プレイヤーの手をそのまま真似た AI を作るのは非常に

困難であるため，ここではプレイヤーの強さだけを抽出した AI を作る．この段階では以下の手順で行い，エージェント AI を作成する．

- (1-1) プレイヤと AI が対戦した記録を取る
- (1-2) その記録をテスター AI が最善手との評価値の差に基づき，評価する
- (1-3) 評価した結果からテスター AI の着手を加減したエージェント AI を生成する

テスターと呼ばれる AI には初級者，中級者よりも強い AI を使用する．詳細なアルゴリズムは 5 章で示す．

### 3.3 多様な戦略を持った AI 群の構成

段階 (2) では遺伝的アルゴリズムを用いてプレイヤーと同レベルの多様な AI 群を構成し，新たな対戦相手として再度人間のプレイヤーと対戦を行う．この段階は (b) 不自然な着手選択を抑制する入出力モデルと (c) プレイヤのレベルに適応するアルゴリズムによって構成され，前者は GA の個体群 (パラメトライズされた思考モデル) に実装し，後者のアルゴリズムとして遺伝的アルゴリズム (パラメータの最適化) を用いる．段階 (1) で作成したエージェント AI と GA の個体を複数回対戦させ，そこから算出された適応度からエージェント AI のレベルを合わせる様に最適化を行う．

また，個体群の AI 同士がそれぞれ違う戦略を持たせるために局面評価関数に使用するオセロの特徴量に着目して適応度計算に組み込む．各個体の特徴量パラメータをお互い離れさせるように GA 進化中に動的に調整することにより，それぞれの AI が異なったパラメータ情報を持つ，つまりはそれぞれが違う場所に着手できることが期待できる．詳細な実験に関しては 6 章で説明する．

## 第4章 オセロ

### 4.1 オセロの概要と，実験に用いる理由

オセロはルールの簡単さから国内でもプレイヤー人口は6000万人ぐらいで他のボードゲームと比べても圧倒的に多い．[11]（将棋：1500万人 囲碁：1000万人 チェス：500万人）

このゲームは二人零和有限確定完全情報であり，完全な先読みが可能である．ルールは8 x 8の盤面上に黒または白の石を配置し，多く石を配置した色のプレイヤーが勝利する．最初に図4.1のように石を配置し，ゲームをスタートする．先攻は黒の石のプレイヤーとなり，縦，横，斜めで相手の色の石を自分の色の石で挟める所を着手できる．着手した後は挟んである相手の石をひっくり返して自分の石にする（図4.2）着手可能点がある場合は必ず着手しなければならず，無い場合はパスをする．両プレイヤーが着手できない場合はゲーム終了となる．

図 4.1: オセロの初期配置

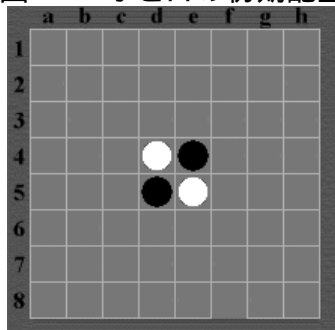
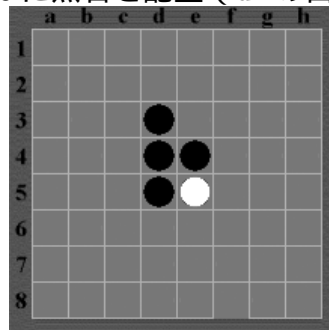


図 4.2: d3 に黒石を配置（d4 の白石を反転）



今回，実験にオセロを用いた理由としてそのプレイヤー人口の多さや単純なルールによる認知度の高さ，そして既存のオセロ AI には LOGISTELLO 以降はどの人間よりも強い評価関数が用意されていることにある．これにより，人間の強さを AI が測れることが可能である．また，強さを調整する研究や自然な着手を行うための研究は比較的少ないのも特徴である．今回使用した AI は Reversi in C# [12] を選び，これに本システムを適用する．

## 4.2 オセロの状態評価関数と特徴量

### 4.2.1 状態評価関数

Reversi in C#の探索法は代表的な木探索である  $\alpha\beta$  法を使用している．そして，その状態評価関数式は線形和によって以下の様に構成されている．

$$ES = w_m(vm_{you} - vm_{opp}) + w_{fr}(fr_{you} - fr_{opp}) + w_s(st_{you} - st_{opp}) + w_{fo} * ff + scr \quad (4.1)$$

$ES$  は評価値， $w_m, w_{fr}, w_s, w_{fo}$  はそれぞれ着手可能点，開放度，確定石，相手がパスしたかに関する重みである．そして， $vm_{you}, vm_{opp}$  は自分，相手の着手可能点の数， $fr_{you}, fr_{opp}$  は自分，相手の開放度の点数， $st_{you}, st_{opp}$  は自分，相手の確定石の数， $ff$  は相手がパスするかどうかの指標（1か0），そして  $scr$  は自分と相手の石の数の差である．オセロの特徴量として特に重要な着手可能点，開放度，確定石については4.2.2，4.2.3，4.2.4節に記述する．

### 4.2.2 着手可能点 (mobility)

オセロの着手可能点はゲームの形勢に大方直結する．オセロは石の数が多の方が勝つと先ほど述べたがゲーム中は石の数の多さはあまり考慮しないほうがいい．これは自分の石を多く取るほど相手の着手可能点が多くなり，下手をすれば自分の着手できる場所が無くなってしまふ，すなわちパスをせざるを得ない状況になりうる．パスの数が多いと劣勢となってしまうため，プレイヤーは次の着手で自分の打てる場所をなるべく多くなるようにする必要がある．そのためにはなるべく石を少なく取る，相手に自分の石を囲ませるなどによって着手可能点を確保しやすく有利になることが多い．ただし，例外もあり図4.3の様な状況では黒が着手可能点が多いのだが4手先では白の完全勝利となる．このことから常に着手可能点が多いほど有利ということではないということも考慮しなければならない．

### 4.2.3 開放度 (frontier)

オセロには開放度理論というものがある．これはオセロの着手において中割りを単純数値化する理論である．中割りとは相手の手をなるべく増やさないようにする手のことであり，一番理想的な中割りは“ ひっくり返すべき石の周りが他の石に囲まれている ”手筋である（図4.4）他にも“ ひっくり返した石がすぐに相手に返されない ”ような手も中割りである（図4.5）

これを単純に数値化するには“ ひっくり返す石の周囲の空きマス数 ”を数えればよい．図4.6は黒番で9個の着手可能点がある．まず，①(d2)の開放度を求めるためにはひっく

り返る石である d3, d4, d5 の周りの空きマスに注目する (空きマスは①(d2) も含める) d3 の周りの空きマスは c2, d2, e2, c3 の 4 マス, d4 は c3 の 1 マス, d4 は c5, e5 の 2 マスである。(図 4.7) これらの合計値は  $4 + 1 + 2 = 7$  となり, これが①の開放度となる. この開放度の数値は小さい程, 次は自分が有利になる展開になりやすい. ここで①以外の開放度を求めると以下の数値となり, この局面では⑥(c6) が一番良い着手となる.

- ② : 5 (1 + 4)
- ③ : 5 (4 + 1)
- ④ : 6 (4 + 2)
- ⑤ : 5 (4 + 1)
- ⑥ : 2
- ⑦ : 4
- ⑧ : 4
- ⑨ : 4

図 4.3: 囲まれたら不利な状況

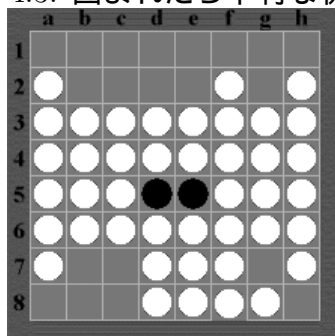


図 4.4: 中割り (反転する石が他の石に囲まれている)

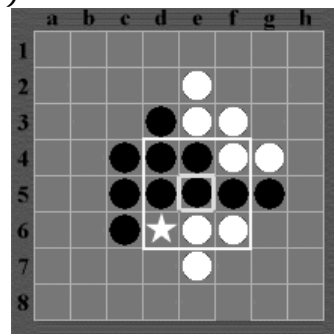


図 4.5: 中割り (反転する石が次に相手に取られない)

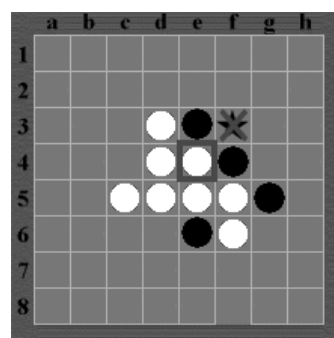
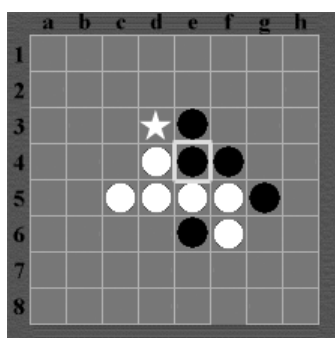




図 4.6: 開放度 (局面の着手可能点)

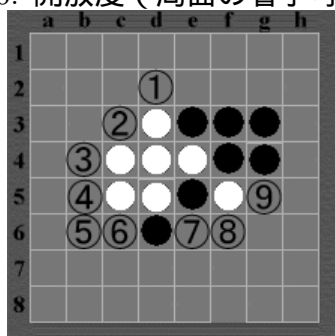
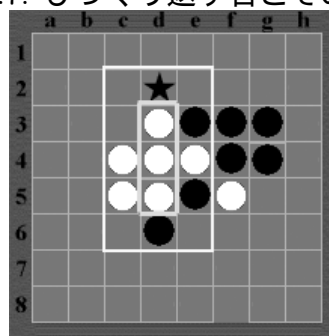


図 4.7: ひっくり返す石とその周囲



#### 4.2.4 確定石 (stable piece)

確定石とは相手に決して取られることの無い石のことである．確定石が増えると相手は決してそれを奪うことができないので勝負において有利な展開になる．その代表的な場所はオセロ盤の4隅であり，比較的分かりやすい位置である（ただし，あくまでも有利になるだけであり隅のみを取ることで自身が勝敗に直結するとは限らない）また，隅以外にも手が進むごとに確定石になる場所があり辺などがその例である．

### 4.3 オセロにおける不自然な手

強いAIを弱くチューニングすると人間にとっては違和感のある不自然な手をうつことがある．人間，特に初心者にとっての不自然な手の定義として様々な解釈があるが，今回は“ 多人数の人間が絶対に打たない手 ”に焦点をおいて考える．実際にアンケートを配って局面を評価してみたところ，初級者にとっての不自然な手，自然な手は以下のものを仮説として立てた．

- 1手先で隅が取られそうな隅の周りのマスに石を置くことは不自然な手
- 盤面の隅のマスが着手可能点なら着手する

オセロによる自然な手はヒューリスティックな評価によって局面評価値に補正值を加える方法を採用した．まず，補正值の決定の為に初級者にとって上記の仮説では無い手を打つ時の評価値の状態を調べた．

図 4.8, 4.9 は仮説に対して，それぞれが隅の周りに着手すると有利，局面は隅に着手すると不利になる局面の一例であり，これらの局面を用いて初級者が仮説とは違う手を出すのかの実験を行った．人間の初級者10人によるアンケート結果では（表 4.1, 4.2），前者の局面はどの隅の周りのマスも8割は打ちたくないと回答した．一方，後者の局面は隅A1に着手すると回答した人は全体の8割であった．他にもアンケートをとった結果，回答者は全て隅の周りは1手先に隅が取られるという理由で着手しない，隅は取れるなら絶

対に着手するという結果となり期待通りの結果にはならなかった．他にも様々な局面を用意したがどれも同様な結果となった．

よって，自然さを表現するモデルは補正値は使わずにヒューリスティックによって隅の周りに着手することで隅が取られる場合は絶対に打たない，または隅が取れるような局面になったら他の評価値がどんなに高くても隅に置くように評価値の書き換えを行う．

図 4.8: 隅の周りが有利な局面

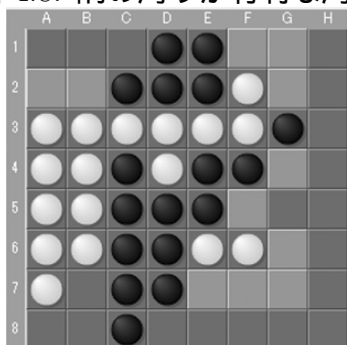


図 4.9: 隅が不利な局面

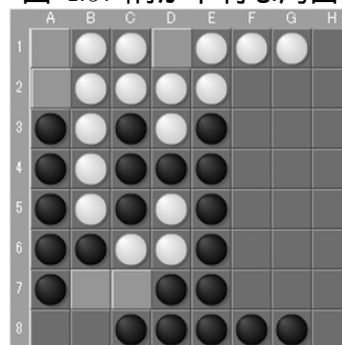


表 4.1: 図 4.8 の人間によるアンケート結果  
(絶対に打たない手)

F1	G1	A2	B2	G2	G4
0	4	8	8	4	1
F5	G6	E7	F7	G7	
0	1	0	1	8	

表 4.2: 図 4.9 の人間によるアンケート結果  
(絶対に打つ手)

A1	D1	A2	B7	C7
8	6	1	0	1

# 第5章 強さの計測と同じレベルのエージェント AI の作成

## 5.1 目標とアプローチ

本章の目標はプレイヤーの棋譜から同じ強さの AI を作ることである。同じレベルのエージェント AI を作成する理由は GA で適応度を算出するためにプレイヤーと同じレベルの AI を用意することで個体群はそれと対戦することにより同じレベルに進化することが期待できるからである。本章では本人とまったく同じ特徴をもった AI ではなく同じ強さだけを真似した AI を生成する理由は人間プレイヤーの特徴は沢山の情報量を有しており、これをそのまま真似するのは非常に時間がかかる上に困難である。よって、着手が不自然でも良い（エージェント AI は GA 内でのみ使うので自然な着手にする必要は無い）ので人間の強さだけを真似した AI を作成する方が簡単で、かつ時間がかからない。今回、学習は GA によるオフライン型の最適化を行っており、人間プレイヤーの代理の AI を用意することにより多くの対戦数をこなすこともできる。

アプローチとして強さの計測、調整を行うためにプレイヤーの着手とテスター AI の最善手との評価値の差（着手エラー）に着目している。強さの計測を行う時にプレイヤーの棋譜をテスター AI が評価し、テスター AI の最善手の評価値とプレイヤーが着手した所のテスター AI の評価値の差を算出する。そして、棋譜をすべて評価した後に着手エラーの平均を求め、それを目標値として評価値を操作、テスター AI をプレイヤーのレベルに合わせ、エージェント AI の実現に図る。テスター AI はプレイヤーの棋譜を評価する立場なので、プレイヤーよりも十分に強い AI でなくてはならない。また、GA のエージェント AI としてテスター AI と同様のモデル、パラメータを用いるので最適化のために速い AI が求められ、着手エラーに一貫性を持たせるために AI の局面評価値は線形性である必要がある（今回使っている局面評価関数は線形和なので問題無い）

今回、テスター AI として以下のものを使用している。このテスター AI の強さは実際に人間のプレイヤー（初心者、または初級者のレベル）と対戦させた結果、表 5.1 の通り、十分に強い結果が出た。この AI 以外にも強いものは公開されているが世界チャンピオンに勝利した LOGISTELLO が公開されて以来、改良したものやより強いものが後発しているので実質人間よりも強い。また、AI の速さは 1 手当たり 2 秒であり、被験者実験で探索深度を浅くすることを考慮すると十分に速いといえる。

- テスター AI の性能

- ソフトウェア名 : Reversi in C#
- 探索法 : minmax 法 ( $\alpha\beta$  法はカットしているため, 全ての着手可能点が正確な値を示していない)
- 深さ : 5
- 相手のパスの重み : 35
- 開放度の重み : 10
- 着手可能点の重み : 5
- 確定石の重み : 50

表 5.1: 人間とテスター AI の対戦結果

人間 (10 人 2 回対戦)	テスター AI
0win	20win

また, 本章の実験の順序として以下の順で行う.

- テスター AI による複数の AI の強さの検証
- 既存方法 (k-best アルゴリズム) による AI の弱さの検証
- 提案手法を用いることによる AI の弱さの検証

## 5.2 複数の AI の強さの検証

### 5.2.1 方法

本節では人間よりも強いテスター AI が複数の強さの異なるプレイヤーの力量を計測することが可能か確かめる. このために用いるテスター AI の最善手の評価値の差である着手エラーはある局面においてプレイヤーが指した手が実際にはテスターからどのくらい実力が離れているかを表している. また, この実験では異なる実力を持つ人間プレイヤーの代理として複数の強さの異なる AI を用いることにより, 強さの計測の妥当性を図る.

強さの検証の方法として以下の順に従い, 1 局当たりの評価値の差の平均, 標準偏差, テスター AI での着手のランクの平均を求める.

1. 人間プレイヤーの代わりとなる  $AI\alpha$  と実際に人間と対戦する AI である  $AI\beta$  の対局を棋譜として保存

2. テスター AI が棋譜上の  $AI\alpha$  の着手を 40 手目まで 1 手 1 手評価を行う

3. 評価することにより着手エラー，そして着手した手の順位を計測する

この手法を用いることにより評価値の差の平均，標準偏差は 0 に近いほど，またランクは 1 位に近いほどその AI はテスター AI の強さに近づくこととなる．また，40 手目までを評価する理由として終盤では勝利（敗北）が確定する時，評価値が急激に大きく（小さく）なるので勝敗が左右される手により評価値の変動が激しくなることからプレイヤー本人の実力を測るには不安定となるからである．

複数の人間を表現するために様々な強さの AI が必要となる．用意した AI は以下のものとなる．

- $AIa$  : Reversi in C# advanced ( $\alpha\beta$  法 探索深度 5)
- $AIb$  : Reversi in C# intermediate ( $\alpha\beta$  法 探索深度 4)
- $AIc$  : UCT (1 手 1 秒)

これらの AI の強さは  $AIa : AIb = 96\text{win} : 4\text{win}$  ,  $AIb : AIc = 95\text{win} : 5\text{win}$  となり，強さの異なる人間の代わりとしての検証が可能である．

実験ではそれぞれの AI が互いに同じ AI と対局を行い，棋譜を 50 枚用意する．そしてテスター AI に着手を評価してもらい着手エラーの平均，標準偏差，着手ランクの平均を調べる．

## 5.2.2 実験結果と考察

図 5.1，図 5.2，図 5.3 は  $AIa$ ， $AIb$ ， $AIc$  が同じ AI と対戦した棋譜を 50 枚用意し，テスター AI に評価した着手エラーの平均，標準偏差，着手ランクそれぞれの 50 枚分の平均をとったヒストグラムである．

結果は一番強い  $AIa$  が平均，標準偏差ともほぼ全ての棋譜が  $AIb$ ， $AIc$  よりも 0 近くに集中している．また，着手ランクでは 9 割近くが 1 位または 2 位の手を平均的に着手している．これにより  $AIa$  は他の AI よりテスターに近い実力を持っていることがわかる．また， $AIb$ ， $AIc$  を比べると評価値の差の平均，標準偏差，着手ランクのどの面においても若干の偏りはあるが全体的に  $AIb$  の棋譜が  $AIc$  よりも高い評価である．よって，テスター AI で計測する手法は強さの指標を示すものとして十分である．

## 5.3 既存方法を用いた AI の検証

### 5.3.1 方法

本節では初級者プレイヤーを相手にするためにテスター AI を弱くすることでプレイヤーの強さに近づける検証を行う．そのために寺田 [13] が将棋プログラムに用いた k-best アル

図 5.1: 評価値の差の平均による評価

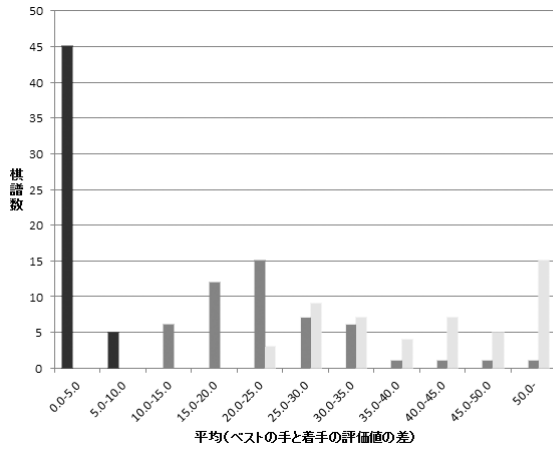


図 5.2: 評価値の差の標準偏差による評価

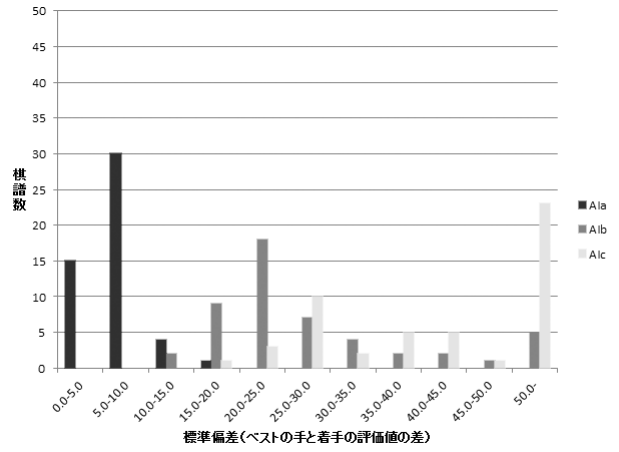
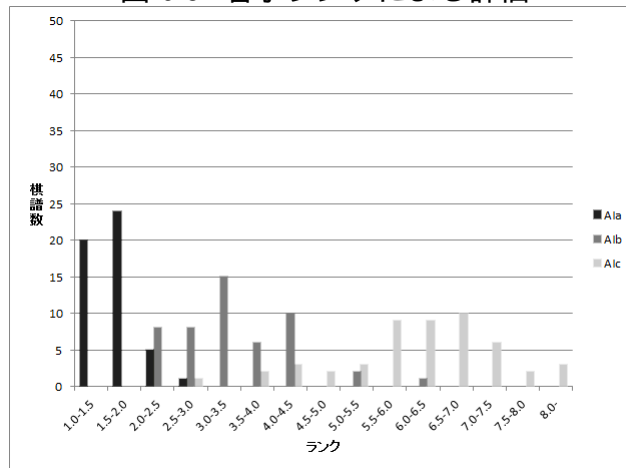


図 5.3: 着手ランクによる評価



ゴリズムをオセロ AI に実装し実験を行う。

k-best アルゴリズムとは強い AI が弱くなりつつも自然な手加減を実現するために使われるアルゴリズムである。k-best アルゴリズムは以下の手順で行う。

1. 高い評価値の順から k 個の手を選択する
2. それらの評価値に  $N(\mu, \sigma)$  の正規分布に基づいた乱数を加える ( $\mu$  は平均,  $\sigma$  は標準偏差)
3. その中から一番評価の高い手を選ぶ

このアルゴリズムを用いることにより明らかに手抜きしている手は選ばれにくくなる。(図 5.4) 寺田はこのアルゴリズムを Bonanza の探索部に適応し、元々強いプログラムから弱く自然な指し手を表現できるに至った。

この k-best アルゴリズムをオセロ AI に適応する際に k の値と正規分布の  $\mu, \sigma$  を決定する必要があり、先ほどテスター AI でプレイヤーの指し手から着手エラーの平均とランクの平均を算出したため、これらを使い値を決定する。

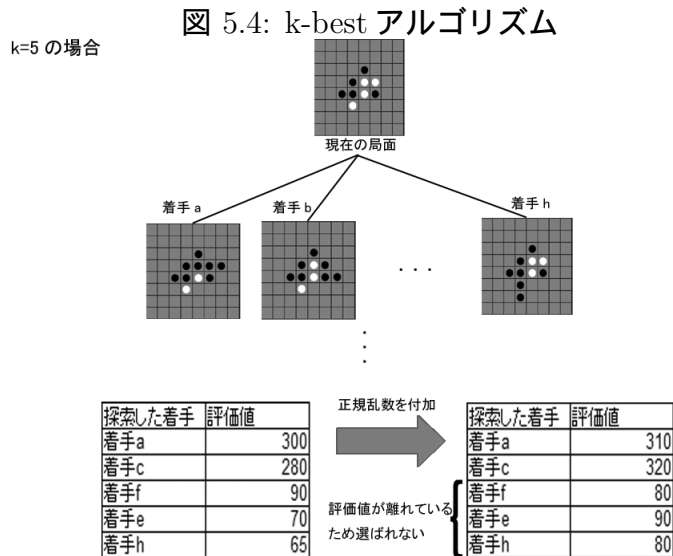
- k : 平均ランクよりも多い値
- $\mu$  : 0
- $\sigma$  : 着手エラーの平均に基づいた値

k の値は複数ある指し手からフィルタを掛けることにより、低いランクの手(明らかに不自然な手)を除外する。ランクの平均をそのまま k の値として使うと k 番目の手よりも低い評価値の手は除かれてしまうので差し手のランクが平均のものよりも高い数値となってしまう。よって、幅広い手に着手できるように k は平均ランクよりも多い値にする必要がある。正規乱数は設定した  $\sigma$  の値に比例して乱数の幅が大きくなるので、 $\mu$  は 0,  $\sigma$  を着手エラーの平均に基づいた数値にすることにより評価値に補正值が付加され、プレイヤーの強さに近づくことが期待できる。

今回はテスター AI に k-best アルゴリズムを実装したものと Reversi in C# の Intermediate ( $\alpha\beta$  法 探索深度 4) を対戦させ、検証を行う。これにより、テスター AI が Intermediate の AI の強さと同じになるか確かめる。

### 5.3.2 実験結果と考察

表 5.2 は k-best アルゴリズムを評価値算出後の深さ 1 に適用したテスター AI を以下の実験条件に従って Intermediate と 100 回対戦した結果である。k,  $\sigma$  の値は 25 枚の Intermediate 同士の対戦の棋譜をテスター AI で評価し、算出したランクの平均と着手エラーの平均のそれぞれの平均値(ランクは 4, 評価値の差は 25.12)を 3 倍した数値である(ランクは k が正の整数にする必要があるため、四捨五入している)



● 実験条件 (k-best アルゴリズム)

- $k : 4 * 3$
- $\mu : 0$
- $\sigma : 25.12 * 3$

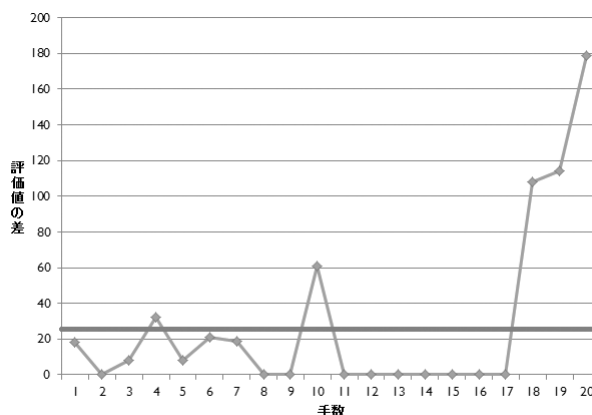
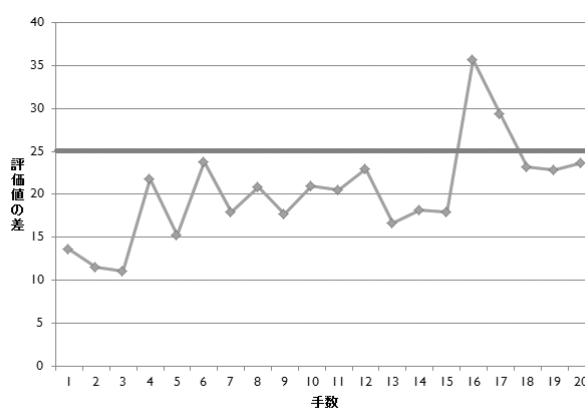
100 試合行った結果、ほぼ互角の強さを持っていることが示されている (表 5.2) ただし、この手法ではプレイヤーの強さと同じものを作れたという根拠は無い。図 5.5 は対戦した棋譜を 25 個抽出し、k-best による着手を元のテスター AI で評価した着手エラーの平均の変化をグラフにしたものである。k-best を適応した AI は手を進めても全体的に評価値の差の平均には届いてはいない。また、図 5.6 は棋譜 1 個分の k-best の着手からテスター AI が算出した評価値の差の推移であり、11 手目から 17 手目にかけてテスターの一番評価の高い手が着手されている。これは正規分布による乱数に依存しているため、他の手の評価値がベストな評価値を超えない、またはベストな手の評価値に大きな乱数が付加されることにより他の手が選ばれなくなるからである。今回は遺伝的アルゴリズムによって同じ強さの AI を作ることが目的であるため、乱数による不安定な強さのエージェントは好ましくないので常に同じ強さの着手を維持できる AI が必要となる。

表 5.2: AI(k-best) と Intermediate の対戦結果 (100 試合)

AI(k-best)	Intermediate
56win	44win



図 5.5: ベストの手と着手の評価の差 (棋譜 25 枚分の平均)      図 5.6: ベストの手と着手の評価の差 (棋譜 1 枚分)



## 5.4 提案手法を用いた AI の検証

### 5.4.1 方法

常に強さを維持する AI を作成するためにテスター AI で評価した着手エラーの平均に沿った手を選ぶ手法を用いる。エージェント AI 作成，着手方法は以下の手順を踏む。

1. テスター AI でプレイヤーの棋譜を評価する
2. 評価の内容は 1 手毎にテスター AI による最善手  $a_{t*}$  と実際の着手  $a_t$  の評価値の差  $v(a_{t*}) - v(a_t)$  を計算する
3. 40 手目 (プレイヤーは 20 着手) までの着手エラーの平均を目標値  $D = \frac{1}{20} \sum_{t=1}^{20} v(a_{t*}) - v(a_t)$  とする
4.  $T$  手目のある局面のここまでの実際の着手と最善手の評価値の差の合計は  $M_{T-1} = \sum_{t=1}^{T-1} v(a_{t*}) - v(a_t)$  と表せる次の着手が  $a$ ，最善手が  $a_{T*}$  としたとき， $T$  手分の最善手と評価値の差の平均は  $D_a = \frac{M_{T-1} + v(a_{T*}) - v(a)}{T}$  とする。次の着手は  $|D_a - D|$  が最小となる  $a$  を選択する

この目標値  $D$  をテスター AI で計測した着手エラーの平均に設定することで常に目標値に沿って着手する，つまりプレイヤーと同じ強さの手を打つことができ，過去の着手との平均を取ることで常に目標値よりも良い手，悪い手と交互にバランスの取れた強さを表現できる。これにより，乱数によって起こる良い手の連続，悪い手の連続が発生しにくくなり GA のエージェントとしての精度が期待できる。

## 5.4.2 実験結果と考察

実際に5種類の様々な手法，強さのAIを用いて提案手法をテスターAIに実装し評価を行う．用意したAIは以下のものである．

- Reversi in C# Intermediate ( $\alpha\beta$ 法 探索深度4)
- Reversi in C# Beginner ( $\alpha\beta$ 法 探索深度3)
- UCT (1手1秒)
- UCT (1手10秒)
- UCT+[14] (UCTに局面評価値を加え改良したモデル)(1手10秒)

それぞれを同じAI同士で対戦し，テスターAIが棋譜を評価，棋譜25枚分の目標値の平均を算出してエージェントAIとの対戦を行った．

表5.3はそれぞれのAIにチューニングしたエージェントAIと5つのAIが対戦(200戦)を行った結果である．エージェントAIは元のAIと互角になっていることから，様々な手法のAIにもこの提案手法は利用できると分かった．よって，様々な人間プレイヤーに対しての妥当性が出てきて同じ強さのエージェントAIとして利用できる．図5.7, 5.8はIntermediate, Beginnerと対戦したエージェントAIの1手1手における評価値の差の推移(棋譜25枚分の平均)である．これらのグラフから平均的に目標値に沿って手を打っていることが分かり，更に1枚分の棋譜における評価値の差の推移(図5.9, 5.10)は目標値に合わせて良い手と悪い手を打っていて，安定した着手が表現できている．

表 5.3: エージェントAIと元のAIとの対戦結果

AIの種類	エージェントAIの目標値	勝ち負け(エージェントAI-元のAI)
$\alpha\beta$ Intermediate	25.12	111-85(0.57)
$\alpha\beta$ Beginner	28.98	96-100(0.49)
UCT1手1秒	36.68	114-87(0.57)
UCT1手10秒	26.51	118-80(0.59)
UCT+1手10秒	21.47	97-99(0.49)

## 5.5 まとめと今後の課題

テスターAIの最善手との着手エラーを強さの指標(目標値)とし，その目標値に沿って着手を決定する提案手法を用いたことにより複数のAIの強さの計測と安定性があり，

図 5.7: ベストの手と着手の評価の差 (Intermediate 棋譜 25 枚分の平均)

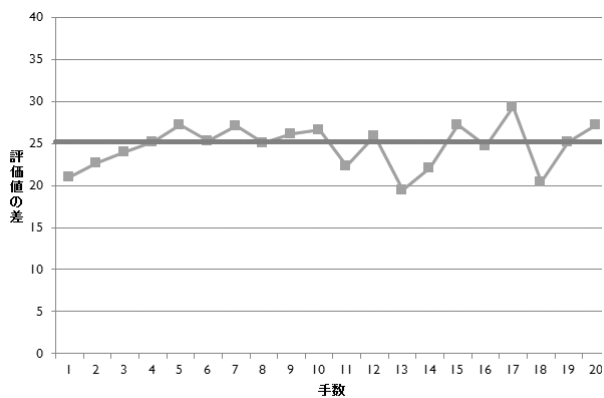


図 5.8: ベストの手と着手の評価の差 (Beginner 棋譜 25 枚分の平均)

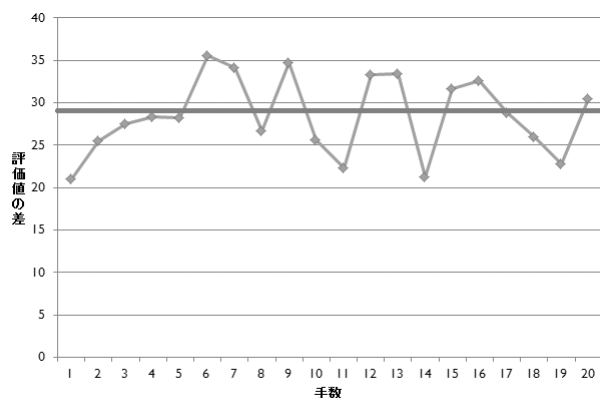


図 5.9: ベストの手と着手の評価の差 (Intermediate 棋譜 1 枚分)

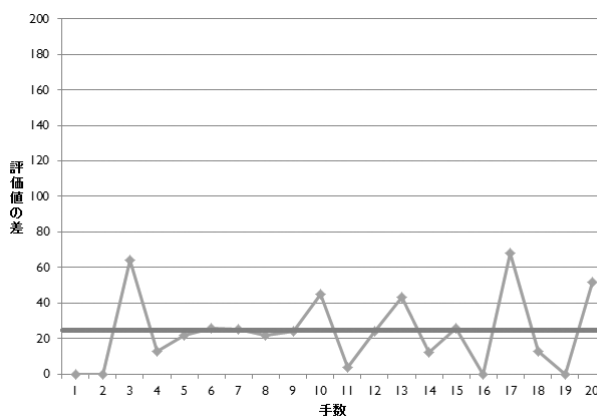
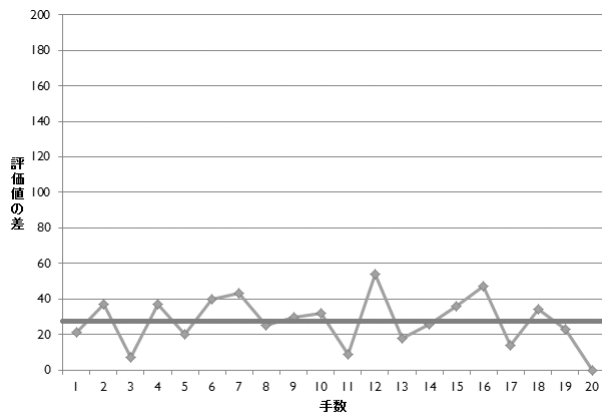


図 5.10: ベストの手と着手の評価の差 (Beginner 棋譜 1 枚分)



様々な手法を用いた AI に対しても同じ強さのエージェント AI が作成できた。これにより、様々な特徴を持つ人間プレイヤーに対してもこの手法が適応でき、強さだけを抽出でき、更に GA の内で個体 AI と対戦することにより同じ強さの AI を複数作ることが期待できる。

今回は複数の棋譜から目標値を定めているが人間相手に強さを測る上では棋譜はモチベーションを考慮すると少ないほうが良い。一方で、棋譜の枚数が 1 枚のみの場合では AI にしろ人間にしろたまたま強く打ってしまった、弱く打ってしまったなど自分の実力とは違う結果が出ることもある。よって、もともとの強さが着手エラー 25 のプレイヤーでもその行動や相手の状況などによって 15 や 35 など離れた数値としてテスターに評価され、強さの計測を誤ることも十分にありえる。

その解決方法として次に対戦する AI には戦況から目標値を操作することで互角に近くするということが挙げられる。そのために必要なものとして  $f(TS, s, t)$  の関数を扱う。 $TS$  は元の評価値、 $s$  は現在の形勢（評価値の状態）、そして  $t$  はターン数であり、あるターン以降一定の評価値に達したら目標値の操作を行うように調整する。

$f(TS, s, t)$  は対戦した評価値の推移から以下のように設定し、Intermediate, Beginner の 2 つのテスト元 AI に対して目標値を  $\pm 10$  に定めたエージェントと対戦した結果は表 5.4, 5.5 となり、互角に近づいてきていることがわかる。これにより、更に実験、調整は必要であるが強さを測り間違えても同じ強さに調整できることが分かった。今後の課題として調整用の関数に更に最適な評価値の範囲、目標値の補正值、そしてより様々な AI による検証実験を通して少ない棋譜数でも互角の勝負ができるようにしたい。

$$f(TS, s, t) = \begin{cases} s \geq 50 : TS + 15(\text{弱くする}) \\ s \leq -160 : TS - 15(\text{強くする}) \end{cases} \quad (5.1)$$

表 5.4: 再調整後 (Intermediate 200 試合)

	エージェント AI 目標値=15.12	Intermediate	エージェント AI 目標値=35.12	
調整前	161win	35win	43win	153win
調整後	99win	98win	89win	107win

表 5.5: 再調整後 (Beginner 200 試合)

	エージェント AI 目標値=18.98	Intermediate	エージェント AI 目標値=38.98	
調整前	162win	38win	98win	100win
調整後	130win	68win	116win	79win

# 第6章 多様なAIを作成するアルゴリズムの実装，検証

## 6.1 目標とアプローチ

本章ではプレイヤーのレベルに適応するアルゴリズムとして遺伝的アルゴリズムを用いることにより，様々な戦略を持たせた複数のAI群の多様性を実現することを目標としている．この実験により人間プレイヤーが対戦を行う度に違う特徴を持った相手と対局ができ，モチベーションを維持できることが期待できる．

GAの個体群については5章で作成したエージェントAIと同じ強さ，またそれぞれの個体AI同士が違う戦略を持たせるように適応度の計算法等を考える必要がある．適応度を求める際，エージェントAIと同じ強さに個体群を調整するために，対戦を複数行い勝率を算出するが，GAの進化中では対戦数に比例して最適化に要する実行時間が長くなることが懸念される．そのため，試行時間を短くしつつ，それぞれの個体とエージェントの対戦勝率を5割，互角に維持するように対戦数を設定しなければならない．

一方，個体群のAI同士がそれぞれ違う戦略を持たせるために局面評価関数に使用する特徴量の係数（パラメータ）に着目している．各個体のパラメータをお互い離れさせるようにGAによって動的に調整することで，それぞれのAIが異なったパラメータ情報を持つ，つまりそれぞれが違う場所に着手できることが期待できる．

今回の実験のアプローチとして以下の手順を踏まえる．

- 全ての個体同士の特徴量パラメータの距離と着手の不一致率の関係
- 勝率に関する適応度と特徴量に関する適応度の推移
- 個体群の特徴量パラメータの3次元空間座標による線形和パラメータ（適応度の重み）の決定
- 最適化された個体AIの特徴の違いの検討

## 6.2 遺伝的アルゴリズム (MGG)

遺伝的アルゴリズム (GA) は John Henry Holland[15] によって提唱され，今日まで大量のデータを扱う組み合わせ問題，非線形問題の最適な近似解を求めるための手法の1つと

して知られている．このアルゴリズムは複数ある解の候補となるデータ（個体）群の間で淘汰，交叉，突然変異を繰り返し最適解へと導いていく．今回使用する GA の世代交代モデルとして Minimal Generation Gap(MGG)[16] を改良し，最も優秀な 2 つの個体を選択する MGG-best2[17]（図 6.1）を扱う．このモデルは以下の手順を踏み，特徴として最適解を求める際に初期収束を防ぐ，解の多様性を保つという利点がある．

1.  $n$  個の個体をランダムに生成する
2. 個体群の中から親を 2 つランダムに選ぶ
3. 選ばれた親個体間で交叉を行い，子個体を  $c$  個生成する
4. 親個体と子個体（ $2+c$  個）すべての適応度を求める
5. それらの中で最良の適応度の個体を 2 つ選ぶ
6. この 2 つの個体を親個体に代わって次世代の個体群に入れる
7. 一定回数行うまで 2 に戻る

### 6.3 BLX- 交叉

今回用いる交叉は実数値交叉である BLX- [18] を用いる．この交叉は親個体の染色体情報の座標と  $\alpha$  の値を元に親の周囲に子個体の座標を新たに生成する．図 6.2 は BLX- を 2 次元空間で表したものである．

親個体の染色体情報の座標を  $(x_1, y_1)$  ,  $(x_2, y_2)$  とすると以下の  $max_{cx}$  ,  $min_{cx}$  ,  $max_{cy}$  ,  $min_{cy}$  が導かれる．

$$max_{cx} = max(x_1, x_2) + \alpha|x_1 - x_2| \quad (6.1)$$

$$min_{cx} = min(x_1, x_2) - \alpha|x_1 - x_2| \quad (6.2)$$

$$max_{cy} = max(y_1, y_2) + \alpha|y_1 - y_2| \quad (6.3)$$

$$min_{cy} = min(y_1, y_2) - \alpha|y_1 - y_2| \quad (6.4)$$

子個体の染色体情報の  $x$  座標は  $max_{cx}$  ,  $min_{cx}$  の間のランダム値， $y$  座標は  $max_{cy}$  ,  $min_{cy}$  の間のランダム値となる．また，実装では座標値に対数（底は 10）を取り交叉を行い，交叉の処理が終了したら指数を用いて元の座標値に戻している． $\alpha$  の値は 0 なら親同士の座標範囲内の子個体しか生成できないが 0 よりも大きい値にすることにより乱数によって座標範囲外の子個体が生まれるため，解の多様性を保つという利点が出てくる（実質，交叉のみで突然変異の要素も担っている）

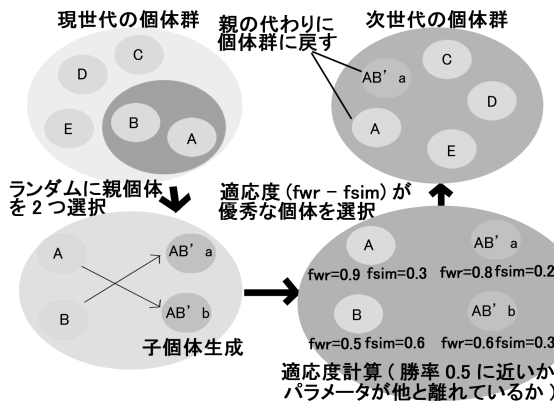


図 6.1: MGG-best2

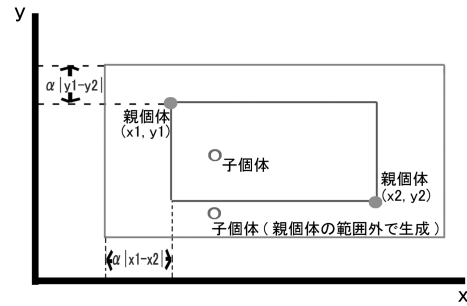


図 6.2: BLX- $\alpha$  交叉

## 6.4 実験

### 6.4.1 方法

プレイヤーと同じ強さでかつ様々な戦略を持たせるように最適化させるために個体 AI はエージェント AI との勝率が 0.5 に近いもの、そして各特徴量パラメータが他の個体とは違う必要がある。式 (6.5) は個体ごとの適応度の計算式である。 $f_{wr}$  は勝率に関する適応度、 $f_{sim}$  は特徴量パラメータの類似に関する適応度、 $\beta (> 0)$  は線形和パラメータ (重み) である。 $f_{wr}$  が大きい ( $\beta$  が小さい) ほど勝率を重視し、 $f_{sim}$  が小さい ( $\beta$  が大きい) ほどパラメータを重視することを意味している。これは前者は勝率 0.5 付近の個体が増加し、後者は勝率 0.5 付近の個体が減少することを意図している。

$$f = f_{wr} - \beta f_{sim} \quad (6.5)$$

勝率に関する適応度は式 (6.6) の式で求める。これは個体がエージェント AI と互角になっているかを評価し、 $wr$  (勝率) が 0.5 になると  $f_{wr}$  の最大値 1.0 に、1.0 か 0.0 になると最小値 0.0 となる。

$$f_{wr} = 4 \times (0.5 - |0.5 - wr|)^2 \quad (6.6)$$

一方、特徴量パラメータの類似に関する適応度は式 (6.7) となる。 $n$  は個体数 (MGG の子個体も含める)、 $(x_i, y_i, z_i)$  は  $i$  番目の個体の特徴量パラメータベクトル (染色体情報) となり、個体同士のパラメータ値の距離が近いほどペナルティとして適応度が高くなる。

$$f_{sim} = \frac{1}{n} \cdot \sum_{j \neq i} \frac{1}{(\log_{10} x_i - \log_{10} x_j)^2 + (\log_{10} y_i - \log_{10} y_j)^2 + (\log_{10} z_i - \log_{10} z_j)^2} \quad (6.7)$$

パラメータに対数を取る理由は局面評価関数が線形和であるため、10 と 100 を比較した場合と 100 と 1000 を比較した場合とでは線形和としての重みとしてはユークリッド距

離に差は変わらないからである．要するに素の数値の差は前者が 90，後者が 900 となり後者の方が大きくなってしまいが対数を取ることで差がどちらも 1 となるので同じ重みを比較していることになる．

今回扱うパラメータとして以下のものを使う．

1. 開放度の重み
2. 着手可能点の重み
3. 確定石の重み
4. 石の差

これら全てを染色体情報として各個体に設定しても良いが、まずは少なめに設定することによって少ない情報量による最適化を実現したいと思う．他に相手がパスしたかのパラメータも今回使用する局面評価関数で扱われているが、これに関しては 1 に含まれるのでこの要素は除き、上記の 4 つに絞り込み実験を行いたい．また、(1, 1, 1, 1) と (1000, 1000, 1000, 1000) のように“本質的に同じ挙動を示すのに遺伝子が異なる”ことを防ぐため、この 4 つの要素のうちどれかは固定値にすることが望ましい．今回は 1, 2, 3 の重みを変動整数値 1 ~ 10000, 5 の重みを固定整数値 100 とする (パラメータの重みは整数型) 本章全ての実験の条件は以下のものを共通とする．

- BLX- の  $\alpha$  値 : 0.5
- 個体数 : 20 個
- MGG での子個体の数 : 10 個
- 個体の深さ : 3
- 染色体情報の要素数 : 3 つ (開放度, 着手可能点, 確定石の重み)
- 要素の初期値 : 100 ~ 1000
- 要素の限界値 : 1 ~ 10000
- 個体 1 つとエージェント AI の試合数 : 20 試合
- GA の繰り返し : 15 サイクル
- エージェント AI のパラメータ : Reversi in C# の Beginner
- エージェント AI の深さ : 2



GA を動かす際に重要なのは最適化に要する時間である．そのため，適応度算出のための試合数を精度を落とさずに少なくする必要がある．今回は個体数が 20 個，子個体の数を 10 個，エージェントとの試合回数 20 試合とすることで最適化に要する試合数は 1 サイクル当たり  $(20/2) \times (2 + 10) \times 20 = 2400$  試合になり，1 試合約 0.3 秒，15 サイクルで行う最適化にはおおよそ 3 時間となる．よって，現実的な時間で最適化が可能である．

### 6.4.2 特徴量パラメータと不一致率の関係

最初に特徴量パラメータ（開放度，着手可能点，確定石の重み）の 3 次元空間における距離と戦略の違いの関係について調査する．各個体 AI 同士のパラメータが離れれば離れるほどそれぞれの戦略が変わる，つまり同じ局面を用意しても違う着手をすることが期待できる．今回は線形和パラメータ  $\beta = 0.01$ ，局面を 300 局面（中級者と Reversi in C# の対局）用意し，GA 適応後のそれぞれの個体の着手の不一致率を調べた．

図 6.3，6.4 は GA を 15 サイクルした後 20 個の個体のうち 2 つをピックアップし，他の個体との特徴量パラメータの距離と着手の不一致率の関係をプロットしたグラフである．距離に関してはパラメータに対数（底は 10）を取り，計算している．2 つのグラフからどちらも距離が個体から離れるほどその個体との不一致率が増加することがわかる．よって，各個体の特徴量パラメータが離れれば離れるほど違う AI として戦略などの特徴が変わることになり，特徴量パラメータの適応度計算が有効であると証明できる．

図 6.3: 特徴量パラメータの距離と不一致率の関係 1

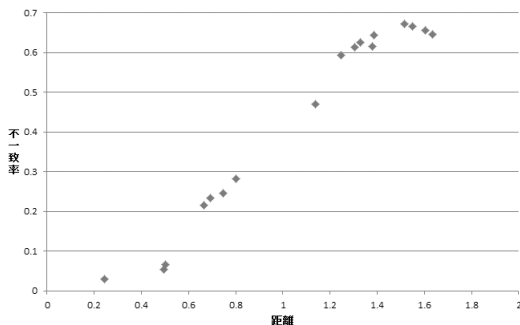
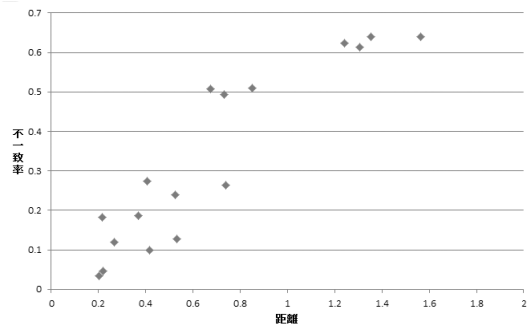


図 6.4: 特徴量パラメータの距離と不一致率の関係 2



### 6.4.3 適応度の推移

本節では線形和パラメータ  $\beta$  を複数用意し， $f_{wr}$  と  $f_{sim}$  のサイクル毎の推移を調査する．6.4.1 で述べた  $\beta$  値を大きくするほど  $f_{sim}$  が重視，勝率が 0.5 から離れていき，小さくなるほど  $f_{wr}$  が重視され勝率が 0.5 に近づいていくかを検証する．線形和パラメータには  $\infty$ ，0.01，0.001，0.0001，0.0 を用意し，適応度はそれぞれ GA を 20 試行した適応度の平均を取る．

図 6.5 は  $f_{wr}$  のサイクル毎の推移，図 6.6 は  $f_{sim}$  のサイクル毎の推移のグラフである．

$f_{wr}$  (図 6.5) は  $\beta$  値が小さくなるほど適応度が 1 に向かって収束し，個体全体がエージェント AI との勝率が 0.5 に近づいていることがわかる．しかし， $\beta = 0.0$  の時は適応度が 1 にはならず  $\beta = 0.0001$  の場合とは変わらない．その理由は進化中のエージェントとの対局数を 20 試合に設定していることによって起こるノイズによるものである．試行の速さを求めるために設けた対局数であるが，適応度計算上の勝率が 0.5 と出た場合でも 100 試合，200 試合と対局数を増やすごとに勝率に違いが生じる．

一方， $f_{sim}$  (図 6.6) は  $\beta$  値が小さくなるほど適応度が大きくなり，個体同士の特徴量パラメータが近づいていく．逆に大きくなるほど適応度が小さくなり，特徴量パラメータが離れるようになっていく．一方で  $\beta$  値が 0.01 と  $+\infty$  の収束の仕方が変わらない．これは  $\beta$  値がある一定以上の値になるとパラメータ同士がこれ以上離れないようになるからである．

図 6.5:  $f_{wr}$  の推移

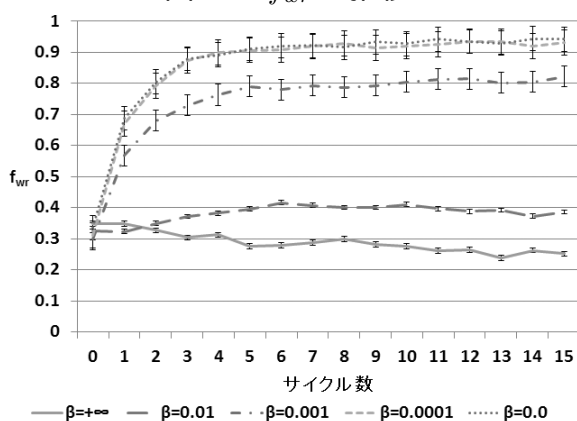


図 6.6:  $f_{sim}$  の推移

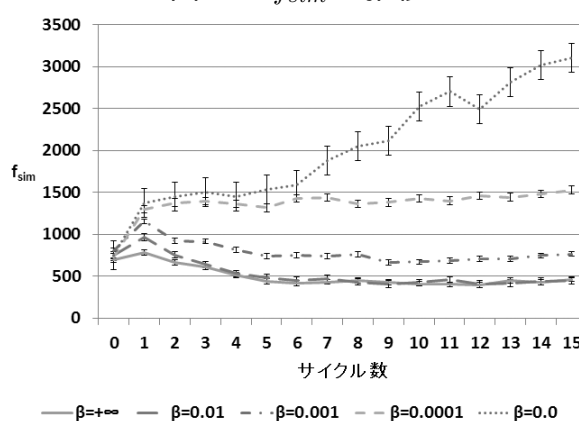


表 6.1 は GA を 1 試行した個体群とエージェント AI を 100 回対戦させた勝率である．引き分けの場合は勝率に 0.5 プラスしている．この表は線形和パラメータ  $\beta$  は 0.01，0.001，0.0001 それぞれの個体の勝率を抽出している．

$\beta = 0.01$  の場合は勝率が全体的に他の  $\beta$  値の個体よりも低くなっている．つまり， $f_{sim}$  の割合が多くなるので勝率が 0.5 から離れている傾向にある． $\beta = 0.001$ ， $\beta = 0.0001$  の場合は  $f_{wr}$  の割合が増えているので勝率が 0.5 に近づいてきている．どちらも 20 個の個体の平均勝率は同じくらいであるが，標準偏差が 0.0001 の方が小さいので，個体の勝率のばらつきが後者が少ない理由で勝率 0.5 近くの個体の数はこちらの方が多くなることとなる．

よって， $\beta$  値の数値が小さいほど個体全体の勝率が 0.5 へと近づき，大きいほど個体同士の特徴量パラメータが離れていることからどちらか一方へと進化していることがわかる．よって， $\beta$  値を調節することにより同じ強さで違う戦略を持つ AI 群ができることが期待できる．

表 6.1: 各個体のエージェントとの対戦勝率 (100 試合)

	$\beta = 0.01$	$\beta = 0.001$	$\beta = 0.0001$
個体 1	0.36	0.37	0.37
個体 2	0.03	0.52	0.45
個体 3	0.38	0.41	0.445
個体 4	0.33	0.59	0.44
個体 5	0.26	0.465	0.49
個体 6	0	0.27	0.51
個体 7	0.07	0.565	0.37
個体 8	0.525	0.4	0.515
個体 9	0.07	0.685	0.38
個体 10	0.165	0.32	0.54
個体 11	0.37	0.41	0.38
個体 12	0.46	0.54	0.41
個体 13	0.09	0.45	0.525
個体 14	0.12	0.485	0.52
個体 15	0.29	0.6	0.565
個体 16	0.08	0.475	0.51
個体 17	0.44	0.595	0.4
個体 18	0.42	0.55	0.5
個体 19	0.27	0.39	0.52
個体 20	0.21	0.26	0.57
平均	0.247	0.4675	0.4705
標準偏差	0.161	0.115	0.067

### 6.4.4 各個体の3次元空間上の分布

この実験では勝率 0.5 付近の個体を抽出して、それらのパラメータのばらつき具合から線形和パラメータを決定する。図 6.7, 6.10 は GA 適応後の  $\beta = 0.01$  の時の個体の特徴量パラメータの 3 次元空間上の分布, 図 6.8, 6.11 は  $\beta = 0.001$  の時の分布, 図 6.9, 6.12 は  $\beta = 0.0001$  の時の分布である。それぞれの分布図は勝率 0.5 付近の目安としてその差が 0.1 以内の個体のみを抽出している。

全体的に  $\beta$  の値が小さくなるほど個体が 1 ヲ所に集中し、大きいほど勝率 0.5 付近の個体の数が減る。プレイヤーのスキル向上のためにはなるべく多様な特徴を持った個体が複数あった方がいいので線形和パラメータはこの中では 0.001 が最も良い値である。

図 6.7:  $\beta = 0.01$  の分布図 1 図 6.8:  $\beta = 0.001$  の分布図 1 図 6.9:  $\beta = 0.0001$  の分布図 1

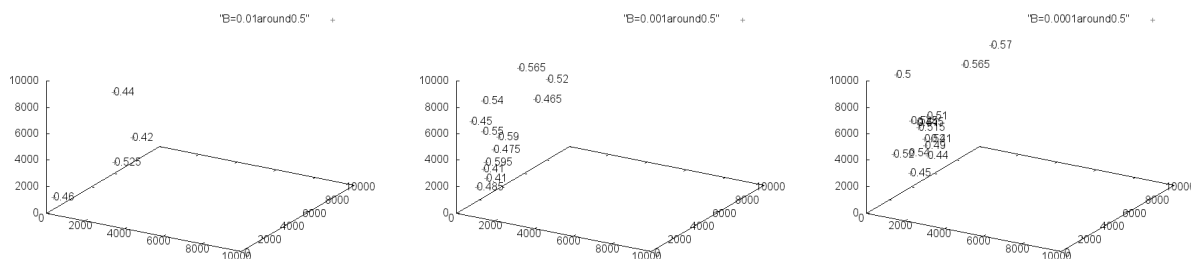
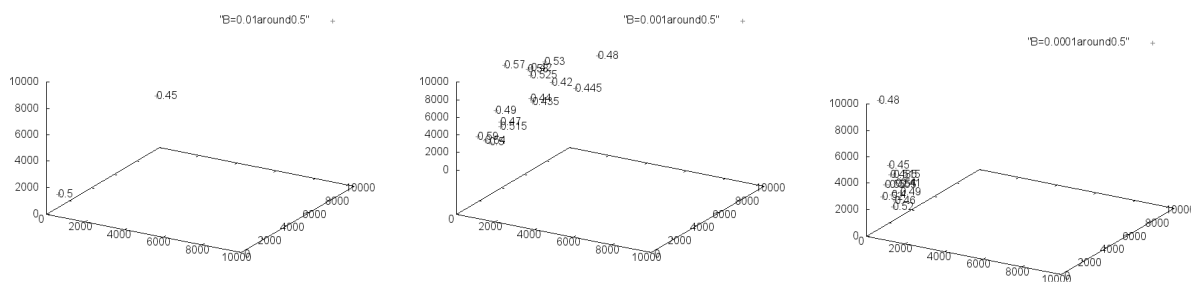


図 6.10:  $\beta = 0.01$  の分布図 2 図 6.11:  $\beta = 0.001$  の分布図 2 図 6.12:  $\beta = 0.0001$  の分布図 2



### 6.4.5 勝率 0.5 付近の個体同士の着手の相違

本節では  $\beta = 0.001$  に焦点を絞り込み、次は実際に最適化された個体群が違う戦略を行うかの検討を行う。検討対象として図 6.13 で示している勝率 0.5 近くであり、以下のアルゴリズムによって特徴量パラメータの異なる 3 つの個体を用意した。

1. 勝率 0.5 付近の個体群の中で最も対数を取ったユークリッド距離的に離れた個体を 2 つ選ぶ
2. (1) で選んだ 2 個体以外で、それらへのユークリッド距離の合計が最も大きい個体を 1 つ選ぶ

これらの特徴量パラメータは表 6.2 となっている。まず、これらの個体における互いの不一致率を求める。不一致率を求めるための局面は 6.4.2 と同様に 300 局面用意し、結果は表 6.3 となった。互いの不一致率はおおよそ 3 割となっており、60 手（片方 30 手）あるオセロの試合の中で片方 9 局面は他の個体とは異なる着手を行うことを意味している。この結果からそれぞれの個体の着手の不一致率は十分違うものが得られている。

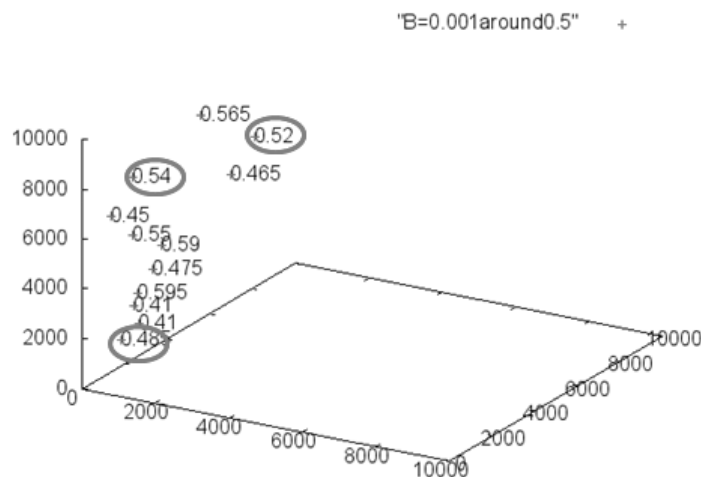


図 6.13: 戦略の異なる 3 つの個体 AI の選択

表 6.2: 3 つの個体の特徴量パラメータ

個体 AI	開放度	着手可能点	確定石	勝率
個体 1	1183	6130	7417	0.52
個体 2	612	1378	8009	0.54
個体 3	692	693	1844	0.485

表 6.3: 3 つの個体の不一致率

	個体 1	個体 2	個体 3
個体 1		0.2533	0.3533
個体 2	0.2533		0.3167
個体 3	0.3533	0.3167	

次にそれぞれ違う着手をする局面において AI がどのような評価を行ったかを調べた。図 6.14 の局面では 3 つの個体がそれぞれ違う場所を選択している。実際に  $\alpha\beta$  法で探索し、3 手先の局面における開放度、着手可能点、確定石のスコアを調べてみると個体 1 は着手

可能点が8と他のものよりも高く評価されている。これは個体1の特徴量パラメータが着手可能点の重みを多く割り振っているためである。この個体は確定石の重みも重視されているが今回は着手可能点を優先していることになる。一方、個体2は他の個体と違い、確定石のスコアが1であり、個体3は開放度が比較的高いスコアとなっている。個体2は確定石重視の特徴量パラメータになっており、個体3は開放度の重みの割合が他の2つの個体と比べて比較的高い値となっている。よって、それぞれの個体が局面上で他の個体とは違う戦略に基づいて評価をしているといえる。

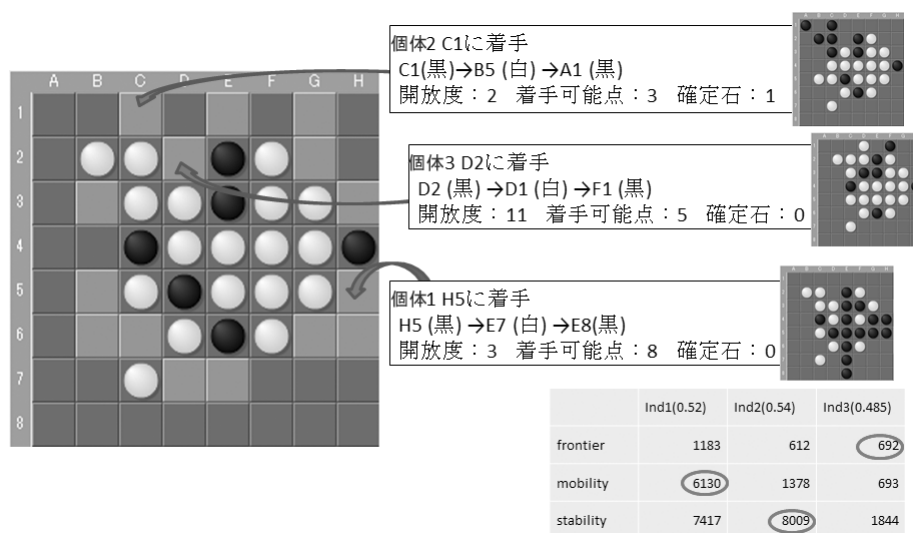


図 6.14: 着手の違い (AI の評価)

#### 6.4.6 ブラインドテストによる多様性の評価

同じレベルで多様な個体ができていることを確認するために人間から見てもこれらの個体が違うものであると分かなければならない。そのために人間によるブラインドテストによってそれぞれの個体の特徴から違うものか同じものかを確認する評価を行う。以下がその実験手順である。

1. 6.4.5 で用意した最適化された個体群とは別に新たに最適化した個体群を GA で生成する
2. その中から勝率 0.5 付近のもの抽出、そこから距離の離れた 3 つの個体 (個体 a, b, c) を更に抽出する
3. それらの個体を対戦 AI として 2 つ組の AI 群を 6 組作る (同じ個体同士 3 組, 違う個体同士 3 組)

4. ランダムの順番で組を出し，対局を行う（対戦相手は中級者である）

5. 1組と対戦した後はペアの個体が同じものか違うものかを評価し，全て終了したら答え合わせをする

中級者が実際に AI 群と対戦した評価と理由は以下の表 6.4 となる．結果は実際のペアの内容と順番の予想と合致し，他にも同様の実験を 4 セット行った結果では 9 割近く製糖したことからこの実験により主観的ではあるが GA によって生成された個体群は人間によって比較，区別が可能であるといえる．

表 6.4: 実際のペアと人間による評価

	実際のペア	中級者	理由
1 ペア目	個体 a, 個体 b	違う	1 つ目は石を多く石を取る，辺に執着 2 つ目は石をあまり取らない，頻繁に中割りする
2 ペア目	個体 a, 個体 c	違う	1 つめは石を多く石を取る，辺に多く石を置く 2 つ目はあまり石を置かない，相手を徐々に隅へと誘導する
3 ペア目	個体 c, 個体 c	同じ	あまり石を取らない，相手を隅近くに打たせるようにする
4 ペア目	個体 b, 個体 b	同じ	あまり石を取らない，隅の周囲に石を打とうとする
5 ペア目	個体 a, 個体 a	同じ	石を頻繁に取る，隅の周囲に石を打とうとする， 徐々に相手の石数に近づいていく
6 ペア目	個体 b, 個体 c	違う	1 つ目は隅の周囲に石を打とうとする，辺の石を伸ばす 2 つ目は相手を徐々に隅のほうに移動させる

## 6.5 まとめ

エージェント AI と同じレベルでかつ様々な特徴，戦略を持たせた AI を複数作るために解の多様性を重視した MGG, BLX- $\alpha$  を用いた．また，適応度計算ではエージェント AI との勝率が 0.5 に近くするように，特徴量パラメータが他の個体と離れるように特化させた．

最初に個体同士の特徴量パラメータベクトルの距離と着手の違い，不一致率の相違の関係を求めるための実験を行った結果，距離が離れるほど不一致率は高く，近くなるほど低くなった．これにより，特徴量パラメータが他と離れるほど着手の違いがより顕著に現れることが証明され，適応度計算で使えることがわかった．

次に線形和パラメータ  $\beta$  による勝率に関する適応度と特徴量パラメータの類似に関する適応度の推移を調べた． $\beta$  値が小さくなるほど勝率の適応度の方が，大きくなるほど特徴量パラメータの適応度の方がより優秀な値へと収束し， $\beta$  値によって特徴量パラメータ

の調整が可能であるとわかった。また、GA 適応後に複数回の対戦による再評価で  $\beta$  値が小さいほど実際の勝率が 5 分に近い個体が増えることもわかった。

3 つの  $\beta$  値から最適化した個体群の特徴量パラメータの 3 次元空間上の分布は  $\beta$  値が高いほど勝率 0.5 の個体が少なく、小さいほど多くはなるが 1カ所に集中する傾向となった。本研究では多様な戦略が期待できる個体を複数必要となるのでこの中では線形和パラメータを 0.001 とした。

そして、 $\beta$  値 0.001 で最適化した個体群から特徴量パラメータの異なる 3 つの個体を選択し、それぞれが同じ局面で違う戦略に基づいて評価を行っていることが探索によるスコアによってわかった。また、中級者による評価では個体それぞれの着手の仕方、戦略から個体が同じか違うかの判別が出来た。これにより、人間でも AI の多様性が判断できることが証明され、この手法を用いて初級者による被験者実験を行う。



# 第7章 被験者実験に基づいたライバル AIの検証

## 7.1 目標とアプローチ

本章では5, 6章で扱った手法, そして4.3節で述べたヒューリスティックによる評価値の書き換えを用いて被験者実験によるシステムの評価を行う。目的として最適化されたAIが被験者と同じレベルになっているのか, 最適化された複数のAIそれぞれの特徴などの違いがわかるのかを調査する。

実験では6.4.6節と同じ要領で以下の手順を踏む。

1. 被験者の強さに最適化された複数のAIを生成する
  - (a) 被験者にオセロの対局をしてもらい, 3枚棋譜を取る (相手は Reversi in C# Beginner レベル ( $\alpha\beta$  法, 探索深度3))
  - (b) 棋譜からエージェントAIを作成, GAで最適化を行い, 同じ強さの個体群を生成する
  - (c) 更に6.4.5節で用いたアルゴリズムを使い, その個体群の中で特徴量パラメータの異なる3つの個体AI(A, B, C)を抽出する
2. 被験者にその3つのAIと対戦して, 評価してもらう
  - (a) 3つのAIを2つ1組のAI群6組(AA, BB, CC, AB, AC, BC)をランダムな順序で用意し, 被験者は1組のそれぞれのAIと1対戦, 計12対戦してもらう
  - (b) 被験者は1ペア毎にアンケートA1枚に答えてもらう (6ペアなので6枚)
  - (c) すべてのペアとの対戦が終了したら, アンケートBに答えてもらう

実験条件は6.4.1節で述べた条件に以下の変更を施し実験を行った。

- 被験者対象の人数: 初級者10人
- 特徴量パラメータの初期値: 1~2000
- GAの繰り返し: 10サイクル

表 7.1: アンケート A

1つ目の AI

1. 自分にとっての強さはどれくらいでしたか

強すぎた やや自分より強い 同じ程度 やや自分より弱い 弱すぎた

2. あからさまな手抜きのような不自然な手はありましたか

しばしばあった 4~6手不自然と感じた 1~3手不自然と感じた まったく感じなかった

3. 2. で少しでも不自然な手があったと感じた人に聞きます. どこが不自然だったかを書いてください

2つ目の AI

1. 自分にとっての強さはどれくらいでしたか

強すぎた やや自分より強い 同じ程度 やや自分より弱い 弱すぎた

2. あからさまな手抜きのような不自然な手はありましたか

しばしばあった 4~6手不自然と感じた 1~3手不自然と感じた まったく感じなかった

3. 2. で少しでも不自然な手があったと感じた人に聞きます. どこが不自然だったかを書いてください

1. この2つの AI は同じか違うかどちらだと感じましたか  
同じ 違う

2. その理由を書いてください

表 7.2: アンケート B

1. 練習相手としてふさわしいと思いましたが  
ふさわしい まあまあふさわしい 普通だった ちょっとふさわしくない ふさわしくない
2. 対戦して楽しいと思いましたが  
とても楽しかった まあまあ楽しかった 普通だった ちょっと物足りなかった つまらな  
かった
3. 更にライバルとして必要なものは何なのでしょう

- 個体と対戦する AI : 目標値に沿って着手するテスター AI (探索深度 4)
- 個体の探索深度 : 2
- 線形和パラメータ : 0.001

## 7.2 実験

### 7.2.1 アンケート結果

図 7.1, 7.2 は 1 人当たり 12 回対戦を行う中で自分と比較しての強さのアンケート結果を総計して表したヒストグラムである。前者は目標値 (着手エラー) がテスター AI に近い人から 5 人, 後者は残る 5 人による総計である。強さの評価は前者が 32% の割合でやや自分より強い, 同じく 32% が同じ程度の強さとし, 半数以上が自分と同じ程度ぐらいと評価した。一方, 後者は全体の 38% がやや自分より強い, それに続いて 28% は同じ程度の強さの評価であり, 前者に比べてやや強めに受けられているがそれでも全体的にはほぼ同じくらいの強さを示している。よって, 強さにおいてはプレイヤーよりも強すぎる, 弱すぎるという評価があったものの大半はほぼ同格かそれ以上と感じられ, 不自然さにおいては初級者から見れば多少は現れるもののほとんどの試合においては違和感が無いということとなる。

図 7.3 は被験者が AI が不自然な手と感じた対局のアンケートを 1 人 12 回対戦, 10 人分の総計を表したグラフである。グラフでは全体の 7 割り近くが着手の不自然さにはまったく感じなかったという結果となった。また, ヒューリスティックを取り入れていない場合

(図 7.4) と比較すると着手の不自然さが少なくなっている為、ヒューリスティックを取り入れることによって不自然さを解消するという検証を得た。

一方で、AI に不自然さが感じられたという理由には以下のものがあった。

- 隅が取られそうなのにわざと取らせる、取らせないようにしていたのに急にやめる
- 隅が取られた後のフォローが粗末だった
- 局面の状況にしては弱い手を打つ
- 相手の辺の確定石を増やす手を打つ
- 石の数を多くとりすぎた
- 序盤と終盤とで強さが変わった

また、図 7.5 は被験者が AI と対戦した勝ち数を表したグラフである。この結果から、8 人は半分ぐらいは勝利したという結果になり、一部の被験者には強めに最適化されたところもあるが、AI が初級者と互角の勝負ができていることがわかった。

図 7.6 は 6 ペアそれぞれが同じ AI か違う AI 同士で構成されているかを判別し、その正答率をヒストグラムで表したものである。この図から 7 人が正答率 0.667、残り 3 人が 0.333 となり、順序がランダムであることを考慮しても、全体の 7 割のみが正答率  $\frac{2}{3}$  で全ての組に正解したものは一人もいない。よって、初級者には用意した AI の違いが判別しづらいという結果となった。これは今回使用した 3 つの特徴量パラメータでは初級者には高度なもので区別するのが難しいことによるものである。

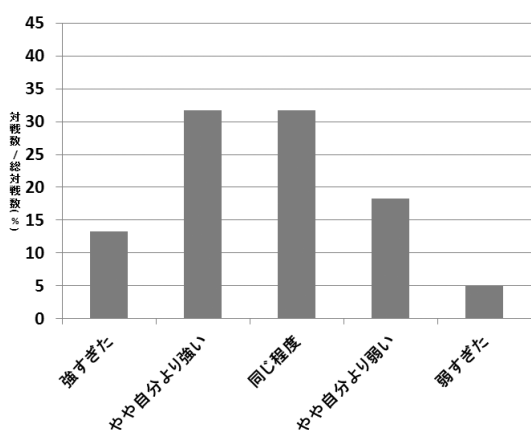


図 7.1: アンケート結果 (強さの評価 上位 5 人)

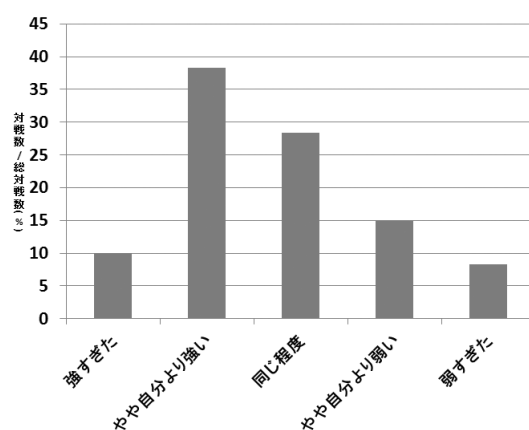


図 7.2: アンケート結果 (強さの評価 下位 5 人)

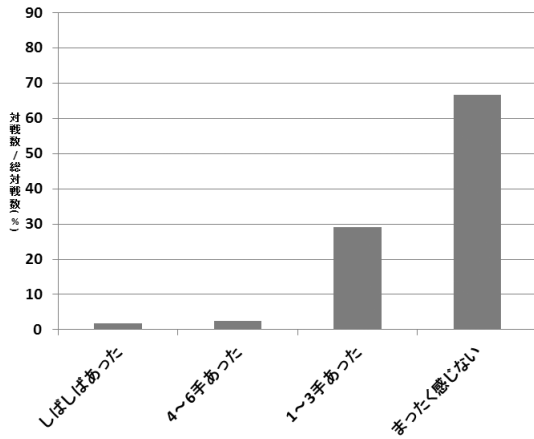


図 7.3: アンケート結果 (不自然さの評価 ヒューリスティックあり)

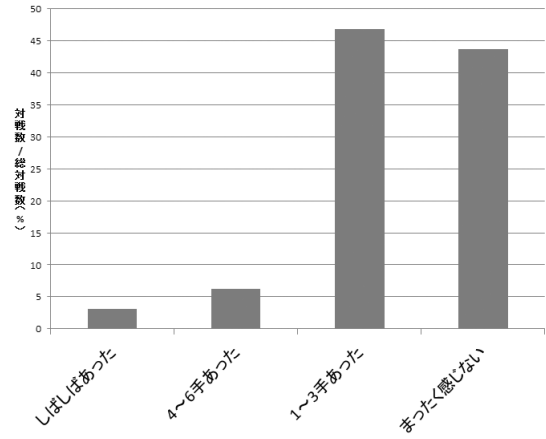


図 7.4: アンケート結果 (不自然さの評価 ヒューリスティックなし)

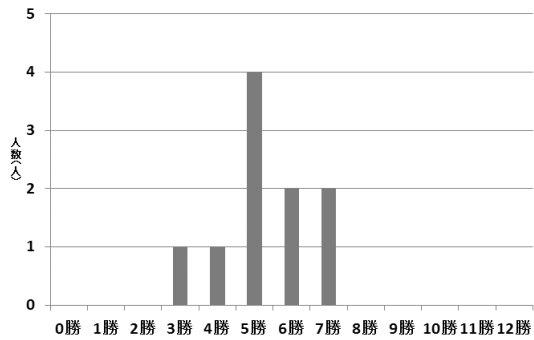


図 7.5: AI との勝率の結果

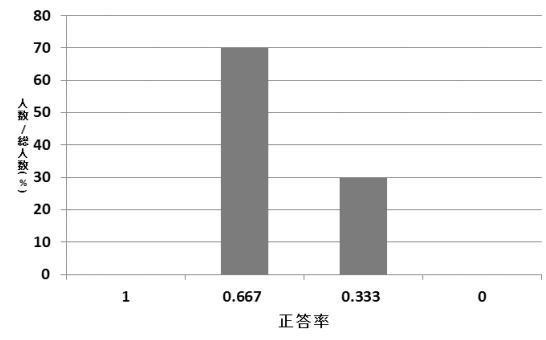


図 7.6: アンケート結果 (多様性の評価)

## 7.3 まとめ

本章では 10 人の被験者に協力をお願いし、学習のために複数の対戦の棋譜を取り、それらの棋譜を用いて GA で最適化を行い生成されたそれぞれが違うステータスを持つ AI 群と対戦を行った。

アンケートによる結果では強さや不自然さ、勝率はプレイヤーと同じぐらいに近づくことが出来た。

しかしながら、多様性の面では初級者に対してあまり良い結果が得られなかった。その対策として今回使用した特徴量パラメータよりもわかりやすいものを染色体情報として追加し、より初級者相手にわかりやすい手を打つようにする必要がある。その他にも、不自然な手があった場合に特に意見の多かった“隅をわざと取らせる”という点にも着目を置き改良を行いたい。

図 7.7, 7.8 はそれぞれ用意した AI が練習相手としてふさわしいか、対戦して楽しかったかをまとめたものである。練習相手として 2 人がふさわしい、7 人がまあまあふさわし

いと答え、対戦してとても楽しかったと答えたのは3人、まあまあ楽しかったと答えたのは6人という結果となった。それぞれ1人は期待した結果にはならなかったが、今後は全員がふさわしさ、楽しさにおいて満足できるように努めたい。

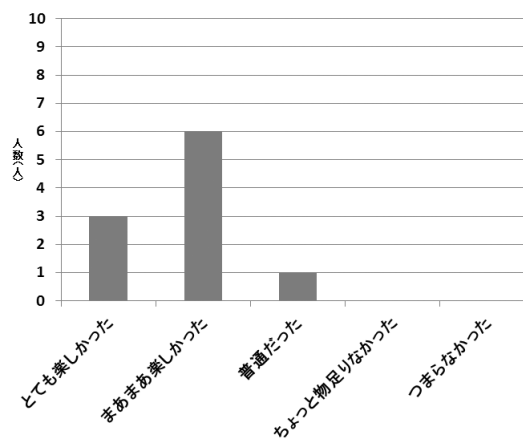
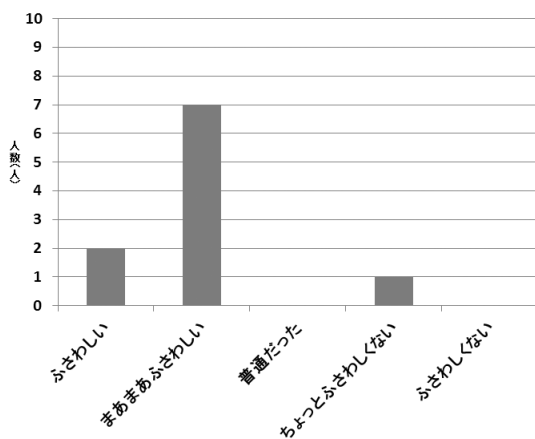


図 7.7: アンケート結果（練習相手としてのふさわしさの評価）

図 7.8: アンケート結果（面白さの評価）

また、更にライバルとして必要なものとして以下の意見をもらった。

- 自分が挑戦したいと思う強さ、適度な強さを持っている
- 自分でもわかるいい手を示す、どのような理由で打ったかの理由を示す
- 自分がどのくらいの強さか示してくれる
- 手抜きを感じさせない
- 隅が取りづらい相手である
- 勝つか負けるかのギリギリの強さを持っている
- 相手を揺さぶる手を打つ

今回のアンケート結果は一部は希望にそぐわないものとなったが“ スキルアップになりいい試合が出来た ”、“ 自分がまるでうまくなったような感じがした ”という意見もあったので更なる改善、改良に努めたい。

## 第8章 おわりに

本研究では初級者のライバルとなりうる AI 群を構成するために遺伝的アルゴリズムを用いたシステムを提案し、オセロに適用した。第1段階ではテスター AI と呼ばれる人間よりも強い AI が棋譜を評価し、最善手との評価値の差を用いて元の人間と同程度の強さのエージェント AI を作成した。第2段階では自然さを考慮したモデルと特徴量パラメータから構成された GA の個体群とエージェント AI を対戦させ、対戦結果が5分になるように、特徴量パラメータの異なりが大きくなるように最適化した。

被験者実験の結果は強さが同じ程度、着手の不自然さが少ない評価を受け、勝率も大半のプレイヤが5分になる結果となりこれらに関しての目的は達成はできた。しかし、特徴、戦略の多様性を感じさせる部分に関しては若干課題が残る結果となった。この原因は今回用いた特徴量パラメータが初級者にはやや高度で判断しづらいためである。そこで、「積極的により多くの石を取るか取らないか」、「辺など、ある一定の場所に打つことを特に好んだり嫌ったりするか」、「縦横のひっくり返しを好むか、斜めのひっくり返しを好むか」など、より分かりやすい表現をするために特徴量パラメータに追加することが今後の課題である。この様に多くのパラメータを追加しても相手の強さに合わせる事が出来るのが遺伝的アルゴリズムを用いたこのシステムの利点でもある。また、従来手法との比較が無い、エージェント AI の着手が固定的であるなどの問題もあり、通常ソフトとの比較も今後の研究課題でもある。

# 謝辞

本研究に際して、様々なご指導を頂きました池田心准教授，飯田弘之教授に深く感謝を申し上げます。

飯田研究室の神谷俊佑氏，橋本隼一氏にも討論の際にご助言頂き，感謝の意を表します。また，実験にご協力いただいた池田・飯田研究室のメンバーには学生生活の面も含め，感謝を申し上げます。



## 参考文献

- [1] C . E . Shannon ”Programming a Computer for Playing Chess”, *Philosophical Magazine*, ser.7, vol.41, no.314, 1950
- [2] Jongyeol Yang, Seungki Min, Chee-Onn Wong, Jongin Kim, and Keechul Jung ”Dynamic Game Level Generation Using On-Line Learning”, *Springer - Lecture Notes in Computer Science*, vol4469, 916-924, 2007
- [3] 加藤 俊明, 鈴木 豪, 小谷 善行, 堤 正義 “ 対戦相手に合わせた評価関数の学習システム ”, 情報処理学会研究報告 . *GI*, [ゲーム情報学] 2000(98), 37-40, 2000-10-26
- [4] 中川 明紀, 柴崎 智哉, 逢坂 翔太, Ruck Thawonmas “ ニューラルネットワークによる格闘ゲーム AI の難易度調整及び行動多様性向上手法 ”, *ゲーム学会和文論文誌*, vol3, no1, 2009
- [5] Pieter Spronck, Marc Ponsen, Ida Sprinkhuizen-Kuyper, Eric Postma ”Adaptive game AI with dynamic scripting”, *Springer - MACHINE LEARNING*, vol63, no3, 2006
- [6] Jean-Marc Alliot and Nicolas Durand ”A genetic algorithm to improve an othello program”, *Artificial Evolution*, 307-319, 1995
- [7] Chuen-Tsai Sun, Ming-Da Wu ”Self-adaptive genetic algorithm learning in game playing”, *Evolutionary Computation, 1995., IEEE International Conference on*, vol2, 814-818, 1995
- [8] Dharm Singh, Chirag S Thaker, Sanjay M Shah ”Quality of State Improvisation Through Evaluation Function Optimization in Genetic Application Learning”, *Emerging Trends in Networks and Computer Communications (ETNCC), 2011 International Conference on*, 93-97, 2011
- [9] Siang Y. Chong, Mei K. Tan, and Jonathon D. White ”Observing the Evolution of Neural Networks Learning to Play the Game of Othello”, *Evolutionary Computation, IEEE Transactions on*, vol9, issue3, 2005
- [10] Darwersi <http://www.arsac.org/olivier/darwersi>

- [11] 長谷川五郎 “ オセロの勝ち方 ”, 河出書房新社出版, 2001
- [12] Reversi in C# <http://www.codeproject.com/Articles/4672/Reversi-in-C>
- [13] 寺田 好彦 “ 人間らしい振る舞いをするコンピュータプレイヤーの設計 ”, 北陸先端科学技術大学院大学修士論文, 2011
- [14] 前原 彰太, 橋本 剛, 小林 康幸 “ 局面評価関数を使う新たなUCT探索法の提案とオセロによる評価 ”, 情報処理学会研究報告 *GI[ゲーム情報学] 2010-GI-24(5)*, 1-5, 2010
- [15] John H. Holland ”Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence (Complex Adaptive Systems)”, published by A Bradford Book, 1992
- [16] 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 *12(5)*, 734-744, 1997
- [17] 池田 心 “ GAによる大域的最適化とその実用に向けた研究 ”, 東京工業大学博士(工学)論文, 2003
- [18] ESHELMAN L. J ”Real-Coded Genetic Algorithms and Interval-Schemata”, *Foundations of Genetic Algorithms 2*, 187-202, 1993