

遺伝的アルゴリズムによる人間のレベルに適応する 多様なオセロ AI の生成

上 田 陽 平^{†1} 池 田 心^{†1}

AI 技術の進展に伴い、多くのゲームで人間より強い AI を作成することが可能になった。一方で、初級中級者プレイヤーにちょうどよい強さの、かつ不自然でない AI を多様に構成することは容易ではなく、課題となっている。本研究では、遺伝的アルゴリズムを用い、ちょうどよい強さと多様さを兼ねた“ライバル AI 群”を構成する手法を提案する。さらにオセロにこれを適用し、被験者実験によって評価を行った。

Human-level Adaptive and Various Othello AIs with Genetic Algorithm

YOHEI UEDA^{†1} and KOKOLO IKEDA^{†1}

Advancement of the AI technology makes stronger AIs than any human players in many games. On the other hand, structure of various Game AIs, which have the same strength as Beginners and Intermediates and act naturally, isn't easy work. I proposed the method for structuring "a group of rival AIs" which has suitable strength for general players and variety of strategies with Genetic Algorithm. In this paper, I applied the system to Othello game and evaluated its performance by an experiment with testees.

1. はじめに

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。競技用に開発された 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もあり、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。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り、自動化が望まれる。

これらの課題を踏まえ、これからの人間とゲーム AI が良い関係を築くために求められる要素として以下のものを提唱したい。

- 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

本研究では主に多様な特徴と強さの調整に着目しており、既存研究でも様々なゲームの種類で行われている¹⁾²⁾

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釣りあったよい相手（広辞苑より）」と定義され、それに基づき以下の3つがライバルの特徴として必要であると考え、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

^{†1} 北陸先端科学技術大学院大学

JAPAN ADVANCED INSTITUTE OF SCIENCE AND TECHNOLOGY

- 対等な相手と感じられること
- 近い力量でありながら、違う特徴・戦略を持つこと

このような 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成する。

(a) プレイヤの強さの計測，模倣

プレイヤの強さを棋譜から抽出し、プレイヤと同じレベルの AI を作成する技術

(b) 不自然な着手選択を抑制する入出力モデル

局面評価から自然な着手を行える AI のモデルを構築する技術

(c) プレイヤのレベルに対応させるアルゴリズム

プレイヤと同レベルでかつそれぞれ異なる振る舞いをする AI を複数作成する技術

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。

本システムは一般のゲームに通用可能であるが、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³⁾ ルールが単純なオセロを対象として実装と評価を行う。また、オセロに関する研究は AI を強くするもの⁴⁾⁵⁾ は数多いものの、自然な手加減を行う研究は少ない。

3. 研究のアプローチ

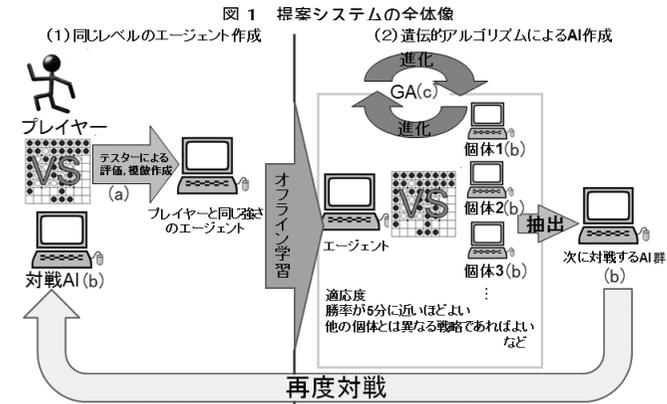
3.1 システム全体のアプローチ

本研究では、人間プレイヤと同程度の強さを持ちつつ、特徴が異なり、不自然ではない 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 1 に示すが、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

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と模倣の要素技術に基づき、プレイヤと同じ力量のエージェントを構成する。プレイヤ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) 記録を評価した結果から、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 4 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 がそれぞれ違う戦略を持つようにするために局面評価関数で使用するオセロの特徴量を適応度計算に組み込む。各個体の特徴量パラメータがお互い離れるように GA で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することが期待できる。詳細な実験に関しては 5 章で説明

する。

4. プレイヤと同じ強さのエージェント AI の作成

初級者プレイヤを相手にするために敢えて弱い AIAI を作る方法としては、探索の深さを制限する方法や評価値に乱数を加える方法など⁶⁾ が一般的である。しかし評価値に単純に乱数を加える方法では、たまたま良い手や悪い手を連続して打ってしまう可能性もあり、その強さは安定しない。本研究でのエージェント AI の役割は、プレイヤの棋譜をもとにプレイヤの強さを再現することであるから、「誤謬度（弱さ）= 最善手の評価値と実際の着手の評価値の差」はより精密に制御することが望ましい。

そこで本研究では、テスター AI がプレイヤの棋譜から誤謬度の分布を推定し、それをエージェント AI として再現するというアプローチをとる。具体的には、テスター AI およびエージェント AI は以下のように構成する。

- 探索法：minmax
- 探索の深さ：5
- 評価関数のパラメータ：Reversi in C# ^{*1}の既存レベル (Expert)
- AI の強さ：初級者との対戦 (20 戦 20 勝)

テスター AI はプレイヤの棋譜から誤謬度の分布を計算し、その平均目標値 μ を定める。エージェント AI は自分の着手の誤謬度の平均が μ に近づくように着手する。具体的な手順は以下の通りである。

- (1) テスター AI でプレイヤの棋譜を評価する
- (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 (プレイヤは 20 着手) までの差の平均を目標値 $\mu = \frac{1}{20} \sum_{t=1}^{20} v(a_{t*}) - v(a_t)$ とする
- (4) T 手目のある局面において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 $\mu_a = \frac{M_{T-1} + v(a_{T*}) - v(a)}{T}$ とする。次の着手は $|\mu_a - \mu|$ が最小となる a を選択する

*1 www.codeproject.com/Articles/4672/Reversi-in-C

4.1 妥当性の検証

本節では、様々な人間プレイヤの代わりとして強さの異なる複数の AI を用意することにより手法の妥当性を検証する。

表 1 は 5 つの AI の棋譜をテスターで評価し、以下の目標値を算出しエージェント AI を生成、元となった AI と対戦した結果である。どのエージェント AI も元の AI とほぼ互角の対戦結果となっていることから同程度の強さであると確認できた。また、これらの元となった AI の詳細は以下に記す。

- $\alpha\beta 1$: $\alpha\beta$ 法, 探索の深さ 4, Reversi in C# Intermediate レベル
- $\alpha\beta 2$: $\alpha\beta$ 法, 探索の深さ 3, Reversi in C# Beginner レベル
- UCT1: UCT, 1 手 1 秒
- UCT2: UCT, 1 手 10 秒
- UCT+⁷⁾: UCB に局面評価を取り入れた UCT, 1 手 10 秒

AI の種類	表 1 エージェント AI と元の AI との対戦結果 テスター AI の目標値	エージェント AI の勝数-負数 (勝率)
$\alpha\beta 1$	25.12	111-85(0.57)
$\alpha\beta 2$	28.98	96-100(0.49)
UCT1	36.68	114-87(0.57)
UCT2	26.51	118-80(0.59)
UCT+	21.47	97-99(0.49)

5. 複数 AI による多様な戦略の最適化

5.1 入出力モデルと特徴量パラメータ

本研究では、AI の入出力モデルには探索の深さが固定の $\alpha\beta$ 探索を用い、更に不自然さを回避するための簡単なヒューリスティックを組み合わせることにする。状態評価関数には、Reversi in C# で用いられている「開放度」「着手可能点数」「確定石数」「石数の差」を 4 つの特徴量を線型和したものをを用い、その重みを GA の個体ごとに持たせることにする。ただし、(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、石数の差については 100 で固定した。

また、自然な着手のモデルについてはアンケートによる予備実験の結果、初級者は盤面上の隅を取る手を非常に好み、取られる手を非常に嫌う傾向が見られるため、前者が打てる場合には必ずその中から選び、後者の場合は必ず回避するヒューリスティックを用いた。なお

アンケートならびに手法に関する詳しい記述は第一著者^{*1}の修士論文を参照されたい。

5.2 遺伝的アルゴリズムについて

今回、ライバル AI 生成の最適化を行うために GA の 1 つである MGG-best2 (図 2) をシステムに組み込む。MGG は以下の手順で行われる⁸⁾

- (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

MGG の利点は解の多様性を維持することなので様々な戦略を生み出す本研究の趣旨に合っている。

5.3 BLX- α 交叉

交叉は実数値交叉である BLX- α を用いる⁹⁾ BLX- α は親個体の染色体情報の座標と α の値を元に親の周囲に子個体の座標を新たに生成する。以下は BLX- α を 2 次元空間で表したものである (図 3)

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。

$$max_{cx} = max(x_1, x_2) + \alpha|x_1 - x_2| \quad (1)$$

$$min_{cx} = min(x_1, x_2) - \alpha|x_1 - x_2| \quad (2)$$

$$max_{cy} = max(y_1, y_2) + \alpha|y_1 - y_2| \quad (3)$$

$$min_{cy} = min(y_1, y_2) - \alpha|y_1 - y_2| \quad (4)$$

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。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) 実験では α の値は 0.5 に設定する。

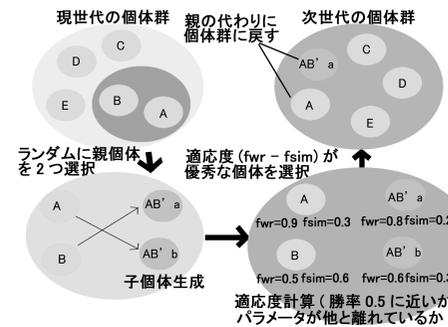


図 2 MGG-best2

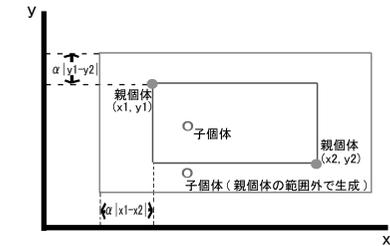


図 3 BLX- α 交叉

5.4 適応度について

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ものであることに着目している。式 (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} が大きいほど、また f_{sim} が小さいほど優秀な個体であることを意味している。

$$f = f_{wr} - \beta f_{sim} \quad (5)$$

勝率に関する適応度は式 (6) 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、 w_r (勝率) が 0.5 になると f_{wr} の最大値 1.0 に、1.0 が 0.0 になると最小値 0.0 となる。

$$f_{wr} = 4 \times (0.5 - |0.5 - w_r|)^2 \quad (6)$$

一方、特徴量パラメータの類似に関する適応度は式 (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っていく。

$$f_{sim} = \frac{1}{n} \cdot \sum_{j \neq i} \frac{1}{(\log_{10} x_i - \log_{10} x_j)^2 + (\log_{10} y_i - \log_{10} y_j)^2 + (\log_{10} z_i - \log_{10} z_j)^2} \quad (7)$$

5.5 実装, 実験について

しかし、適応度計算のための対戦の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、それぞれの個体のエージェント AI に対

*1 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

する勝率が 5 割になるように対戦数を設定しなければならない．全ての実験の条件は以下のものを共通とする．

- 個体数：20 個
- MGG での子個体の数：10 個
- 個体の探索深さ：3
- 特徴量パラメータの初期値：100 ~ 1000
- 特徴量パラメータの限界値：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
- エージェント AI の特徴量パラメータ：Reversi in C#の既存レベル (Beginner)
- エージェント AI の探索深さ：2

最適化に要する試合数は 1 サイクル当たり $(20/2) \times (2 + 10) \times 20 = 2400$ 試合であり，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．

実験は適応度の推移， β 値による個体の特徴量パラメータの分布，そして個体群の着手の相違についての検証を行った．

5.6 適応度の推移と β 値による差異

まず，適応度の推移に関しては線形和パラメータを 5 つ用意しそれぞれの違いを調べた． f_{wr} (図 4) は β 値が小さくなるほど適応度が 1 に向かって収束し，個体全体のエージェント AI との勝率が 0.5 に近づいていることが分かる．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 AI 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 5) は β 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 β 値が 0.01 と $+\infty$ の収束の仕方は変わらない．これは β 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．

これらの結果から， β 値によってパラメータの類似の適応度が制御できていると確認できた．また， β 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，勝率重視かパラメータ重視かの一方へと進化し

ていることが分かる．よって， β 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．

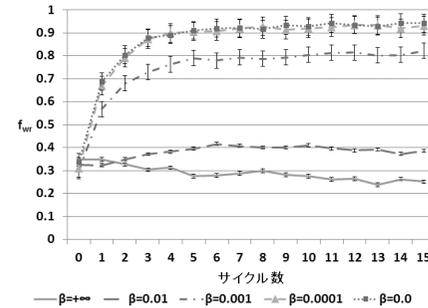


図 4 f_{wr} の推移

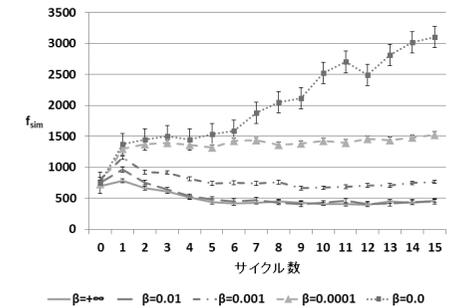


図 5 f_{sim} の推移

5.7 個体の特徴量パラメータの分布と β 値による差異

図 6, 7, 8 は GA の最適化が終了した後の個体群から勝率 0.5 付近の個体を抽出，それぞれの特徴量パラメータの 3 次元空間上の分布を表したものである．勝率 0.5 付近の目安として差が 0.1 以内のものに限定している．全体的に β の値が小さくなるほど個体が 1 か所に集中し，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．

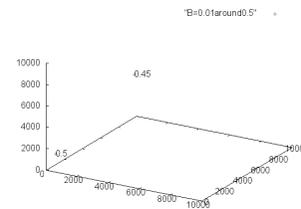


図 6 $\beta = 0.01$ の分布図

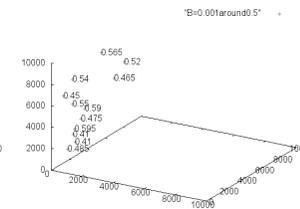


図 7 $\beta = 0.001$ の分布図

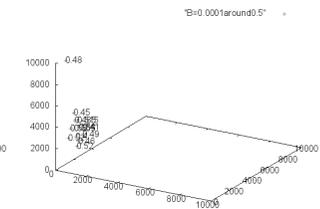


図 8 $\beta = 0.0001$ の分布図

5.8 個体群の着手の相違

実際に GA で最適化された個体群が違う戦略、着手を行っているかを調べるために $\beta = 0.001$ と設定して個体を 3 つ選んだ。(図 9) これらの個体は以下のアルゴリズムによって互いの距離がなるべく離れ、特徴量パラメータの異なるものを選んでいる。

- (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

表 2 は個体の戦略パラメータであり、表 3 は 300 個の局面を用意して 3 つの個体における互いの着手の不一致率を表している。互いの不一致率はおおよそ 3 割となっており、60 手の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。

個体 AI	開放度重み	着手可能点重み	確定石重み	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 2 3 つの個体の特徴量パラメータ

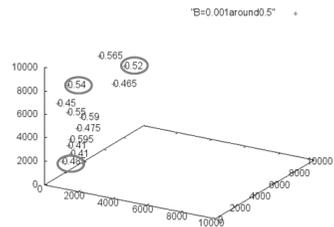


図 9 戦略の異なる 3 つの個体 AI の選択

	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	

表 3 3 つの個体の不一致率

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 10

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 は着手可能点のスコアが 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。これは個体 2 が確定石重視の戦略パラメータとなっており、個体 3 は開放度の重みの割合が他の 2 つの個体と比べて高い値となっているからである。よって、それぞれの個体が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。

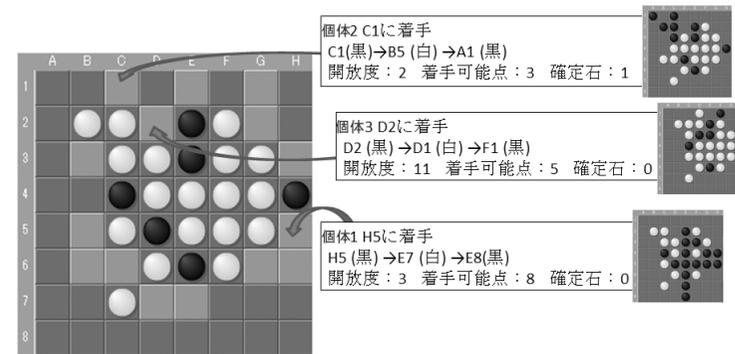


図 10 着手の違い (AI の評価)

5.9 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るの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らなければならない。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の評価を行う。以下がその実験手順である。

- (1) 5.8 節で用意し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) ランダムな順番で組を出し、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 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、この実験により 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。

	実際のペア	表 4 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 つ目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 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
5 ペア目	個体 a, 個体 a	同じ	
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. 被験者実験による提案システムの性能評価

本章では 4, 5 章で扱った手法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 を判別できるかを調査する。

実験手順として、5.9 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 ($\alpha\beta$, 探索深さ 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 5.8 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 1 枚答えてもらう (6 ペアなので 6 枚)
- (c) アンケート内容は AI が自分と比べて強いものか, 不自然な着手がどの程度あったのか, 2 つの AI が同じものか違うものかを問うものとなっている

実験条件は 5.5 節に以下の変更をし実験を行った。

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

図 11, 12 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, 着手の不自然さを 10 人分総計し表したグラフである。強さの評価は 34% の割合でやや自分より強い, 30% が同じ程度の強さと評価し, 不自然さの評価は 65% がまったく感じないという結果となった。よって, 強さにおいてはプレイヤと強すぎる, 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。また, 図 13 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たことがわかった。

図 14 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。

7. ま と め

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

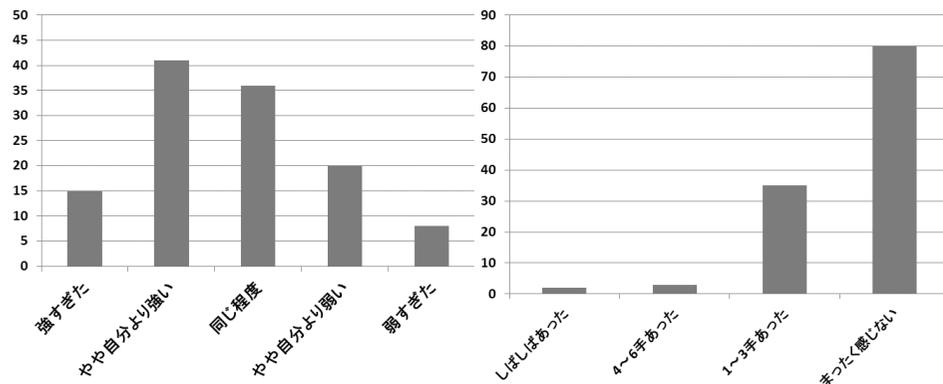


図 11 アンケート結果（強さの評価）

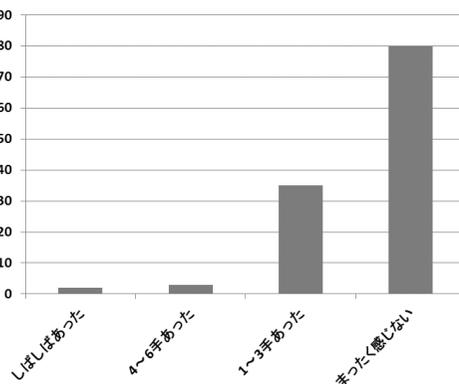


図 12 アンケート結果（不自然さの評価）

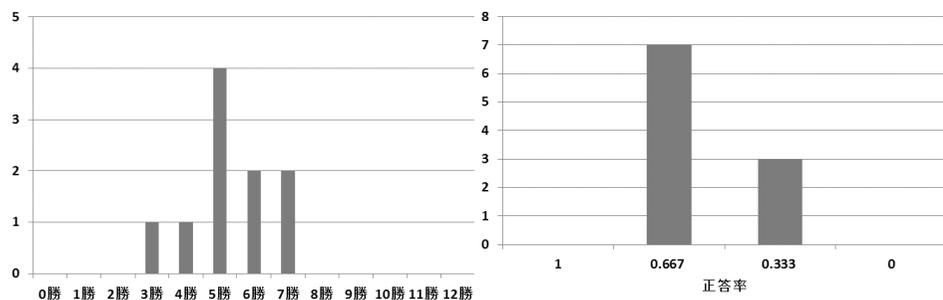


図 13 AI との勝率の結果

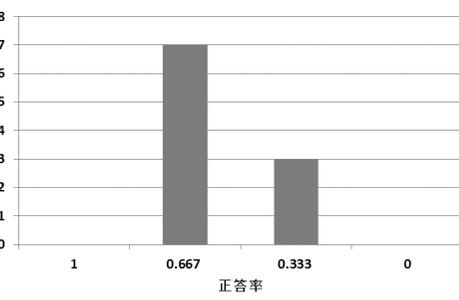


図 14 アンケート結果（多様性の評価）

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ーが5分になる結果となり、これらに関する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。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。

参考文献

- 1) 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
- 2) Pieter Spronck, Marc Ponsen, Ida Sprinkhuizen-Kuyper, Eric Postma ”Adaptive game AI with dynamic scripting”, Springer - MACHINE LEARNING, vol63, no3, 2006
- 3) 長谷川五郎 “ オセロの勝ち方 ”, 河出書房新社出版, 2001
- 4) Jean-Marc Alliot and Nicolas Durand ”A genetic algorithm to improve an othello program”, Artificial Evolution, 307-319, 1995
- 5) 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
- 6) 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
- 7) 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
- 8) 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
- 9) ESHELMAN L. J ”Real-Coded Genetic Algorithms and Interval-Schemata”, Foundations of Genetic Algorithms 2, 187-202, 1993