2009－GI－21 (10)
2009／3／9

GAとTD(λ)学習の組み合わせによるゲーム局面評価パラメータの調整

矢野友貴[†1] 柴田剛志[†2] 横山大作[†3]
田浦健次朗[†2] 近山隆[†4]

ゲームの局面評価パラメータの調整において, 進化的アルゴリズム (EA) 及び強化学習を単独で用いた事例は数多く存在するが, これらの手法を組み合わせて用いた事例は極めて少ない. 本研究では, 遺伝的アルゴリズム (GA) 及び TD(λ) 学習をハイブリッド GA の考え方に基づいて組み合わせる手法を提案する. 本手法をオセロの局面評価パラメータの調整に用い, GA と TD(λ) 学習のバランスを取るパラメータの最適値を実験的に決定した. そして, 本手法と従来手法を比較した結果, 本手法によって調整されたパラメータは従来手法によって調整されたパラメータに対して 54.3%～69.5%の勝率を記録した.

Adjustment of game stage evaluation parameters with combining GA and TD(λ) learning

Yuki YANO,[†1] Takeshi SHIBATA,[†2] Daisaku YOKOYAMA,[†3]
Kenjiro TAURA[†2] and Takashi CHIKAYAMA[†4]

While evolutionary algorithms(EA) and reinforcement learning(RL) are widely used to adjust game stage evaluation parameters, there are few examples to combine these two methods. In this study, we propose a new method combining genetic algorithms(GA) and TD(λ) learning on the basis of hybrid GA. We applied the method to the adjustment of Othello stage evaluation parameters and decided optimal parameters of balancing GA with TD(λ) learning by experiments. A player with parameters tuned with the proposed method showed winning rates of between 54.3% to 69.5% against players tuned with published methods.

1. はじめに

コンピュータゲームプレイヤにおいて, 局面の有利不利を正確に判断する局面評価関数は非常に重要な要素の一つである. 正確な局面評価関数を作成するためには, それを構成するパラメータをうまく調整することが必要となる. この調整を人手で行うことは, 扱うゲームに対する熟練した知識と膨大な労力が要求されるため, 極めて困難である. そのため近年では, 機械学習を用いた局面評価パラメータの自動調整手法の研究が多くなされている. 局面評価パラメータを自動調整する方法は大きく分けて教師あり学習と教師なし学習の 2 つに大別できる. 教師あり学習はゲームの棋譜といった教師データを用いて局面評価パラメータを調整する手法であり, 現在多くのゲームの局面評価関数の学習に用いられている. 一方, 教師なし学習は事前のデータを必要としない学習であり, 棋譜を集めることが困難な未知のゲームに対しても適用可能である反面, 局面評価関数の強さは多くの場合教師あり学習によって調整されたものに劣る.

教師なし学習の代表的な手法として, 進化的アルゴリズム (EA) 及び強化学習[1] がある. EA は大域探索によって最適解を広く探索する能力に長けている反面, 計算コストが大きくなるという短所を有する. 一方, 強化学習は EA とは逆に短時間での学習が可能であるが, 探索が局所的となってしまう. ゲームの局面評価パラメータの調整において, これら 2 つの手法を単独で用いた例は数多く存在しており, また 2 つの手法を組み合わせて用いることでよりよい結果が得られる可能性も示唆されている[2]. しかし, 実際に組み合わせ

†1 東京大学工学部電子情報工学科
Department of Information and Communication Engineering, The University of Tokyo
†2 東京大学大学院情報理工学系研究科
Graduate School of Information Science and Technology, The University of Tokyo
†3 東京大学 IRT 研究機構
IRT Research Initiative, The University of Tokyo
†4 東京大学大学院工学系研究科
Graduate School of Engineering, The University of Tokyo

て局面評価パラメータの調整を行ったという報告は極めて少ない.

一方, EA と局所探索を組み合わせる手法は広く研究されており, 特に EA の一種である遺伝的アルゴリズム (GA)[3),4)] と局所探索を組み合わせる手法はハイブリッド GA[5)] と呼ばれ, 多種多様な分野で高い成果をあげている.

本稿では, 強化学習の一種である TD(λ) 学習[1)] を局所探索とみなすことで, GA と TD(λ) 学習をハイブリッド GA の観点から組み合わせる手法を提案し, 各手法間のバランスを取るためのパラメータ調整及び従来手法との比較実験を行った. 本手法に対してオセロの局面評価関数を例に実験を行ったところ, 同一のリソースを用いた従来手法に比べて 54.3%〜69.5%の勝率を得ることに成功した.

本論文では以降, 2 章において関連研究を紹介し, 3 章で本研究の手法, 4 章で実験結果について説明し, 5 章にて結論と今後の課題を述べる.

2. 関連研究

2.1 強化学習

強化学習[1)] は未知の環境下で試行錯誤を繰り返し, 最適な方策を学習する手法である. ゲームプレイヤにおいて, その行動は局面評価関数によって決定されるため, 方策の学習は局面評価パラメータの調整を意味する. 強化学習は学習を高速で行うことが可能である長所を有する反面, 探索が局所的なため大域的最適解を得ることは困難である.

2.1.1 TD(λ) 学習

TD(λ) 学習[1)] は TD 誤差と呼ばれる隣接する二つの状態間の推定価値の差及び報酬によって定義される値を用いて方策を調整していく手法である. ゲームにおいて, 状態の推定価値とは局面の有利さを表す指標であり, 局面評価関数の出力に相当する. また, 報酬は勝ち負けによって与えられる. t ステップ目の状態を s_t, 推定価値を $V(s_t)$, 報酬を r としたとき, TD 誤差は次式によって定義される.

$$\delta = r + \gamma V(s_{t+1}) - V(s_t) \qquad (1)$$

ここで γ は割引率を表し, [0.0,1.0] の範囲の値をとる. TD(λ) 学習ではこの誤差を 0 に近づけることを目標にパラメータの調整を行っていく. また, TD(λ) 学習の大きな特徴として, TD 誤差による修正を直前の状態だけでなくエピソード中に訪問したすべての状態に伝搬させることが挙げられる. これは適格度トレースとよばれ, その伝搬率はトレース減衰パラメータ λ によって制御される. λ も γ 同様 [0.0,1.0] の範囲の値をとる.

2.2 進化的アルゴリズム (EA)

進化的アルゴリズム (EA) とは生物の進化を模した最適化アルゴリズムである. EA は焼きなまし法 (SA) やタブー探索と同様, メタヒューリスティックアルゴリズムであり, 適用可能な問題の範囲が非常に広く, 多種多様な問題に対して用いられる. ゲームにおける進化的アルゴリズムの代表的な例としては共進化を用いた学習[2)] が挙げられる. 進化的アルゴリズムは強化学習に比べて広く解の探索を行うため, 多くの場合強化学習によって調整されたパラメータよりも良い解を得ることが可能である. しかし, 解が収束するまでに時間がかかり, 計算コストが大きいという欠点がある.

2.2.1 遺伝的アルゴリズム (GA)

遺伝的アルゴリズム (GA)[3),4)] は EA の一種で, 解の集団に対して遺伝子操作を適用していくことで最適解を探していく手法である. 遺伝子操作は, 集団の中から相対的によい個体を選択的に次世代に残す「選択」と, 集団内の個体に各種変更を加える「交叉・突然変異」の 2 つのフェーズからなる.

2.2.2 ハイブリッド GA

GA は解空間を広く探索する能力に長けている一方で, 局所的な最適解を求めることを苦手としている. そのため, この弱点を克服する方法として GA と局所探索を組み合わせたハイブリッド GA が多く提案されている[5)]. 最も単純なハイブリッド GA では, GA と局所探索を交互に実行することによって解の探索を行う. ハイブリッド GA は局所探索によって最適解の候補を絞り込むため, 短期間で良い解を得られる. しかし一方で, 最適解の候補を絞り込むことは同時に探索範囲を狭める結果となるため, GA に比べて局所解に陥りやすい.

ハイブリッド GA の性能を引き出すには, GA と局所探索をどのように組み合わせるかが非常に重要となる. これに対して, Mathias 等はハイブリッド GA の局所探索を頻度, 確率, 期間の 3 つのパラメータで制御する Non-Adaptive Hybrid Algorithms[6)] を提案した. NAHGA で用いられる 3 つのパラメータはそれぞれ, 頻度は GA を何世代行うごとに局所探索を行うか, 確率は局所探索を行う際にどれほどの確率で適応するか, 期間は局所探索をどれほどの時間行うか, ということを意味している. NAHGA ではこれらのパラメータを固定の値として扱っているが, これを集団の状態によって動的に変化させる Self-Adaptive Hy-

brid Algorthms[7] という手法も提案されている.

NAHGA で用いられるパラメータの他に, 局所探索の結果をどのように個体に反映させるかという問題がある. ハイブリッド GA において, 局所探索の結果の反映方法には Lamarckian 方式と Boldwin 方式[8] の 2 つが存在する. 前者の方式は局所探索の結果を個体そのものに反映させるというものである. 一方, 後者は個体には変更を加えずに適応度のみを変更する. 一般的に, Larmackian 方式は収束は早いものの, 個体そのものに手を加えるために局所解に陥りやすいという欠点を有しているため, Baldwin 方式を用いる方が好ましいとされている. しかし一方で, 適用する割合を調整することで Lamarckian 方式が高い性能を示すという報告もある.[9]

2.3 ゲームの局面評価パラメータへの応用

EA と強化学習を組み合わせてゲームの局面評価パラメータの調整を行った事例として次の 2 つがある.

Singer はオセロの局面評価パラメータの調整に GA と強化学習を交互に用い実験を行った[10]. この実験は 3ヶ月の計算時間を要し, 最終的にだいたい中級者程度の強さを得たと報告されている.

Kim 等はオセロの局面評価パラメータに対して, 初めに TD 学習によって個体を生成し, それら個体に対して EA を適用するという実験を行った[11]. この実験によって得られた局面評価パラメータは, CEC 2006 Othello competition[12] において優勝を納めている.

3. 提案手法

2.2.2 節で述べたように, GA と局所探索を組み合わせたハイブリッド GA は多くの分野で成功を納めている. 本研究では, TD(λ) 学習を局所探索とみなすことによってハイブリッド GA に適用する方法を提案する.

3.1 TD(λ) 学習を用いたハイブリッド GA

提案手法の流れは図 1 のようになる. 本研究ではゲームの局面評価パラメータの調整を目的としているため, 集団内の個体は局面評価パラメータとなる. 本手法は 2.2.2 節で述べた NAHGA をベースとしており, TD(λ) 学習を確率的に行う点が大きな特徴である. なお, NAHGA では確率の他に頻度, 期間というパラメータを用いると述べたが, 頻度は確率と統合して扱うことが可能なため本手法では頻度については考えない. また, 期間に関しては 4 章で議論をする. 図 1 を見ると分かるように, 本手法は大きく分けて GA フェーズと TD フェーズの 2 つに分けることができる. 次に, 各フェーズでの動作の詳細について述べる.

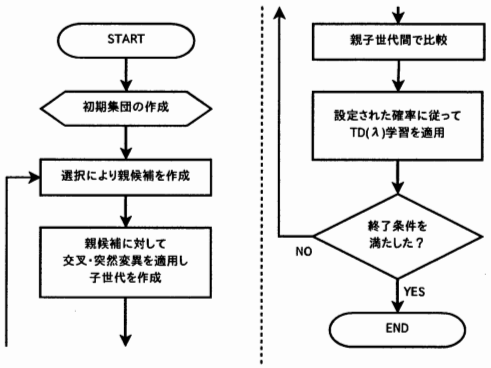

図 1 提案手法のフローチャート

3.2 GA フェーズ

GA フェーズでは次のような操作が行われる. まず現在の集団 (親世代) から親候補を選択する. 次に親候補に対して交叉, 突然変異を施し子世代を作成する. 最後に親世代と子世代の間で比較を行い, より強い個体を次世代に残す. ここで用いられる選択, 交叉, 突然変異, 比較の 4 つの遺伝子操作の詳細は以下の通りである.

選択は 2 つの個体を対戦させ, その勝率に応じた確率を用いて行う. ある個体の勝率が win であった場合, その個体が選ばれる確率は次のように求める.

$$prob = \frac{1}{1 + \exp\{-12 * (win - 0.5)\}} \tag{2}$$

交叉は一様交叉を用いて行う. これは, 任意の 2 個体の局面評価パラメータに対して, 対応するパラメータをランダムに交換することを意味する.

突然変異は次式のように各パラメータの標準偏差を用いて行う.

$$\sigma(i) = \sqrt{\frac{\sum_{j=1}^{M}(\theta_j(i) - avg_i)^2}{M - 1}} \tag{3}$$

$$\theta(i) = \theta(i) + RND(-1.0 \ldots 1.0) \times \sigma(i) \tag{4}$$

ここで, i は局面評価パラメータのインデックス, $\theta(i)$ は i 番目の局面評価パラメータ, $\theta_j(i)$ は集団内の j 番目の個体の i 番目の局面評価パラメータ, avg_i は i 番目の局面評価パラメータの集団全体での平均, M は集団の個体数, $RND(-1.0 \ldots 1.0)$ は [-1.0, 1.0] の一様乱数を表す.

比較は選択と同じように 2 個体を対戦させて行うが, 比較では勝率から確率を求めるのではなく, 単純

に勝率の高い個体を選択する．また前述のように，選択が親世代同士で対戦を行うのに対し，比較は親世代と子世代の間で対戦を行う．

3.3 TD フェーズ

TD フェーズではランダムなペアを作成し，それに対して一定の確率で TD(λ) 学習を施す．これ以降 TD 学習を行う確率のことを「TD 確率」，また，1 回の TD(λ) 学習の試合数を「TD 期間」と呼ぶこととする．なお，TD(λ) 学習によってパラメータが変化した個体は GA フェーズで行った比較のようなことはせずに，そのまま集団に戻す．

3.3.1 スパースな特徴に対する学習率

1 試合の間に出現する特徴が全体の数%しかないスパースな特徴を学習する場合，すべての特徴に対して統一した学習率を用いる方法ではうまく学習することができない．特に各特徴が出現する確率に大きな隔たりがある場合，一部の出現しやすい特徴を優遇することとなる．例えば，特徴 A が毎試合出現し，特徴 B が 100 試合毎に出現する場合，特徴 A は特徴 B に比べて 100 倍の頻度でパラメータの更新が起こることとなる．また，特徴 B は出現するタイミングが特徴 A に比べて遅くなるため，初回学習時の学習率が異なる値となってしまう．そこで，このようなスパースな特徴に対する学習率として (5) 式のような各特徴ごとに独立な学習率を提案する．

$$\alpha[i] = \alpha_0 \times \frac{N_0 + 1}{N_0 + count[i]^{1.1}} \tag{5}$$

ここで，i は特徴のインデックス，$\alpha[i]$ はその特徴に対応した学習率，$count[i]$ はその特徴の出現回数を表し，α_0 及び N_0 はそれぞれ学習率の初期値，減少率を決定するパラメータである．

TD(λ) 学習を用いた予備実験の結果，従来の学習率に比べて提案した学習率はスパースな特徴の学習に対してより高い勝率を記録した．

4. 実　　験

本手法を評価するために，オセロの局面評価パラメータの調整を用いて，最初に初期値，TD 確率，TD 期間について実験したのち，他の手法との比較を行った．

4.1 実験条件

実験には，ネットワークに広域分散したクラスタ群である inTrigger[13] を用いた．GA フェーズの選択・比較及び TD フェーズは各個体が独立に対戦を行うため，複数の CPU を用いて並列に実行が可能である．そのため，実験では個体の管理及び個体全体を必要とする計算を行うマスタプロセスと各個体で独立な部分の計算を行うワーカプロセスを用意した．計算には Intel Xeon E5410(2.33GHz) を搭載したクラスタを用い，1 つのマスタプロセスおよび 64 個のワーカプロセスの計 65 並列で計算を行った．

4.1.1 局面評価パラメータ

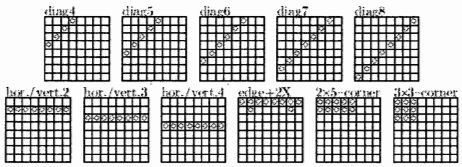

図 2 Logistelo で用いられる特徴

本実験では Logistello[14),15)] で用いられている局面評価パラメータを用いて学習を行った．Logistello では図 2 に示されたパターン及び Parity(空きマスの偶奇) の計 12 種類のパターンを局面評価に用いており，そのパラメータの総数は対称性も考慮して 113879 個となる．ある局面 s での最終的な評価値 $V(s)$ は局面に出現したパラメータの線形和 $f(s)$ を (6) 式のようなシグモイド関数を用いて [0.0,1.0] に正規化して求めた．

$$V(s) = \frac{1}{1 + e^{-f(s)}} \tag{6}$$

4.1.2 TD(λ) 学習の学習率

Logistello で用いられる局面評価パラメータは，1 試合の間に出現する特徴は全体のうちの 1%未満と非常にスパースであるため，3.3.1 節で述べた学習率を用いた．本実験では $\alpha_0 = 0.01$, $N_0 = 100$ とした．

4.1.3 ゲームの進行度と行動

選択・比較・学習の際に行われる対戦では序盤，中盤，終盤を次のように定め，各進行度に応じて行動を変化させた．

双方合わせて 10 手打つまでを序盤とした．序盤では，単純に勝率を求める選択・比較はあらかじめ決められた手を打つものとし，TD 学習では完全ランダムな手を打つものとした．また，学習の際，序盤ではパラメータの調整は行わないこととした．

残り空きマスが 6 マス以下になるまでを中盤とした．中盤では選択・比較・学習とも深さ 1 の探索で最良の手を決めるものとし，学習は中盤でのみパラメー

タの調整を行った．また，学習では ε-greedy 方策に基づき，最良の手以外に $\varepsilon = 0.05$ の確率でランダムな手を打つものとした．

残り空きマスが 6 マス以下のときを終盤とした．終盤では各フェーズともに $\alpha\beta$ 探索によって終局までを読み切り行動を決定した．

4.1.4 その他パラメータ

本実験で用いられる固定のパラメータは表 1 の通りとした．

表 1 各種パラメータ

パラメータ	値
個体数	128
交叉率	80%
突然変異率	1%
トレース減衰パラメータ (λ)	0.7
割引率 (γ)	1.0

4.1.5 強さの評価方法

本実験では，個体の集団からトーナメントによって集団を代表する個体を一つ選び，その個体の強さを評価することで各世代の集団の強さを評価した．具体的な強さの指標として，オープンソースのオセロプログラムである Zebra[16] に対する勝率を用いた．なお，Zebra との対戦は 4.1.3 節で述べた選択・比較と同じ方法で行い，Zebra もそれに従った行動を取らせた．

4.2 各種実験の概要とその結果

4.2.1 初期値に関する実験

3 章でも述べたように，本手法は GA と TD(λ) 学習という異なる手法を混合して用いる．そのため，2 つの手法によるパラメータの調整幅のバランスをとるような初期値を適切に設定する必要がある．そこで，± 1 から $\pm 10^{-5}$ まで初期値の範囲のオーダーを 1 ずつ変化させた計 6 つの初期値を用いて 500 世代の学習を各初期値で 10 回行い，その性能を比較した．このとき，TD 確率は 10%，TD 期間は 1000 試合とし，計算には 1 試行あたり約 5 時間を要した．10 回の試行の結果の平均値及び標準誤差を図示したものを 図 3 に示す．

図 3 より，± 1 から $\pm 10^{-2}$ にかけて勝率が大きく向上したのち，わずかに減少していることが分かる．初期値の範囲が大きい値の場合に勝率が伸び悩む原因としては，1 回の TD(λ) 学習によるパラメータの調整幅に対して初期値の範囲が大きすぎるために，ほとんど GA 主体の学習になってしまっていることが考えられる．これは，4.2.2 節で示す GA のみの場合の結果と類似している点からも推測される．一方，初期値の範囲が小さい場合には逆に TD(λ) 学習によるパラメータ

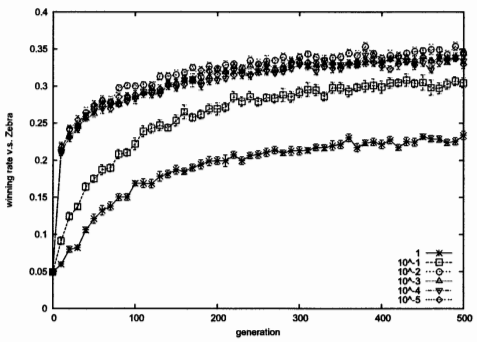

図 3 初期値と勝率の推移の関係

調整の影響が相対的に大きくなるが，初期値が大きい場合ほど大きな性能低下は起きていない．

以上の結果より，以降の実験では初期値の範囲として $\pm 10^{-2}$ を用いることとした．

4.2.2 TD 確率に関する実験

TD 確率は集団内の個体に対して学習を実際に行う確率である．本手法は 2.2.2 節で述べた Lamarckian 方式によるハイブリッド GA であるが，この方式では確率の値による性能への影響が大きくなるため，この確率を適切に設定することはハイブリッド GA の性能を引き出す上で非常に重要となる．このため，TD 確率の大まかな最適値の得るために TD 確率を 0%, 5%, 10%, 20%, 50%, 100%と変化させ，4.2.1 節と同様の方法を用いて性能の比較を行ったところ，図 4 のような結果を得た．なお，TD 確率 0%とは GA のみの場合を意味する．また，1 試行あたりの計算時間は TD 確率が増えるに従って増加し，0%では約 4 時間，100%では約 6 時間の時間を要した．

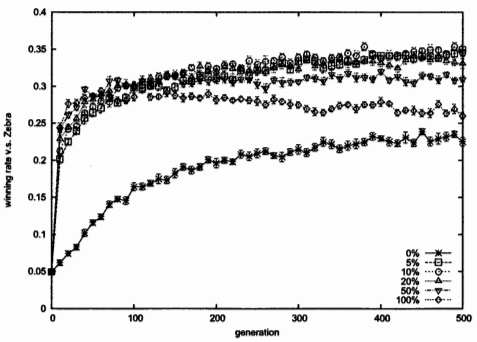

図 4 TD 確率と勝率の推移の関係

図 4 を見ると，TD 確率を 0%にした場合の結果が

他の結果のそれと大きく異なり, 前半での勝率の伸びが緩やかであることが分かる. これは, TD 確率 0%がまったく TD(λ) 学習を用いないために, パラメータの調整に時間がかかることが原因である. 一方, TD 確率を 50%, 100%と大きな値にした場合には, 早い世代での勝率の伸びはいいが, 100 世代以降で勝率が伸びていない. この原因は, 本手法が Larmackian 方式を用いているために, TD 確率が大きい場合には局所解に落ち込んでしまうためである. しかし, TD 確率が 100%の場合は後半で勝率の低下が起きており, これは前述の理由では説明ができない. この原因は不明であるが, 本手法が個体の適応度として Zebra に対する勝率ではなく, 個々の個体同士での勝率を用いているため, その勝率の不一致が結果に影響している可能性が考えられる. この実験でよい結果を出しているものは, TD 確率が 5%, 10%, 20%といった小さな値に設定した場合であり, これらの結果から TD 確率は 10%前後の値に設定することが望ましいと考えられる.

以上の結果より, 以降の実験では TD 確率として 10%を用いることとした.

4.2.3 TD 期間に関する実験

TD 期間の勝率への影響を調べるために, TD 期間を 10 試合, 100 試合, 200 試合, 500 試合, 1000 試合, 2000 試合と長くしていった場合の勝率の変化を 4.2.1, 4.2.2 節と同様の方法を用いて実験を行った. その結果, 図 5 を得た.

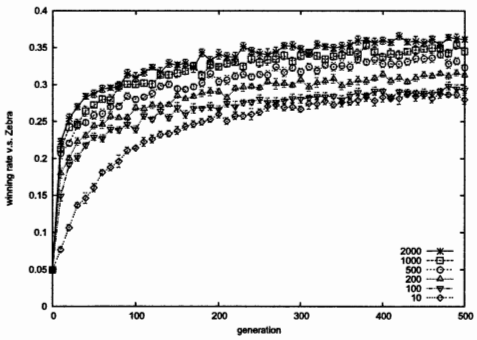

図 5 TD 期間と勝率の推移の関係

図 5 より TD 期間を長くしていくことで勝率が増加していっていることが読み取れる. これは, 4.1.2 節で述べたように, 本実験で用いている特徴が非常にスパースであり, 一回の学習で調整されるパラメータが非常に限られているため, TD 期間を長くすることによって学習中に調整されるパラメータの種類が増え, その結果性能の向上につながったと推測される. 本実験では TD 期間として 2000 試合までしか行っていないが, より長くすることで性能の向上が期待できる. しかし, TD 期間を長くすることは計算時間の増加を招く.

各 TD 期間での計算時間を検討した結果, クラスタ間の通信時間のロス及び計算時間の短縮を考慮して, TD 期間には 1000 試合を用いることとした.

4.2.4 各種手法との比較実験

本手法と従来手法を用いて性能の評価を行った. 従来手法として次の 4 つの手法を用いた.

- GA のみ, つまり TD 確率 0%(GA only)
- TD(λ) 学習のみ (TD only)
- Singer の手法[10] を模した毎世代すべての個体に学習を施す手法, つまり TD 確率 100%(Singer's method)
- Kim 等の手法[11] を模した最初の 10 世代を TD(λ) 学習のみ, その後 GA のみと切り替える手法 (Kim's method)

この 4 つの手法の初期値は TD only のみ 0.0, その他は本手法と同じとした. また, TD only は本手法と計算時間を合わせるために, 1 世代での学習期間を 2000 試合とした. 実験方法は前節と同じように, 500 世代の学習を 10 回行い, その平均及び標準誤差を求めた. 結果は図 6 のようになった. 1 試行あたりの計算時間は, 本手法が約 5 時間, GA only 及び TD only が約 4 時間, Singer's method が約 6 時間, Kim's method が約 3 時間となった.

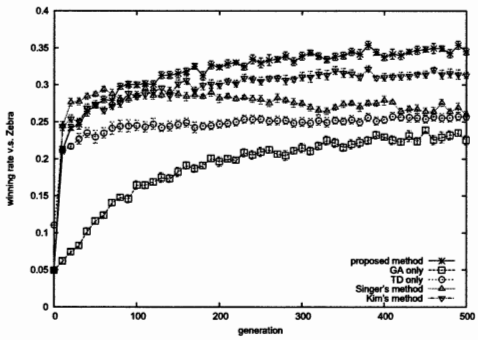

図 6 各手法と勝率の推移の関係

図 6 より, 本手法が他の手法に比べて Zebra に対して高い勝率を記録していることが分かる. また, 本手法を GA only 及び TD only と比べると, 初めの世代では TD only と同じような勝率の伸びを, その後は

GA only と同じような勝率の伸びを見せており, 2 つの手法の特徴をうまく取り入れられていることが読み取れる. しかし一方で, Singer's method の勝率は 60 世代あたりから下降しており, すべての手法を同じ世代数で評価することは適切でないと考えられる. そこで, 累計世代数が同一になるようにした場合の勝率の変化を調べた.

累計世代数は世代数 × 反復回数と定義するとし, 1024 世代を 1 回, 512 世代を 2 回, 256 世代を 4 回, ···, 16 世代を 64 回のように実験を行った. 各世代での強さの評価は, 各試行で得られた個体すべてを用いてトーナメントを 50 回行い, 各トーナメントで優勝した個体の Zebra に対する勝率を平均することで行った. ここで, TD only だけは GA を用いておらず, 各学習が独立に行われるため, トーナメントには集団を代表する個体ではなく集団すべて (つまり 128 個体) を用いるものとした. 実験の結果, 図 7 を得た.

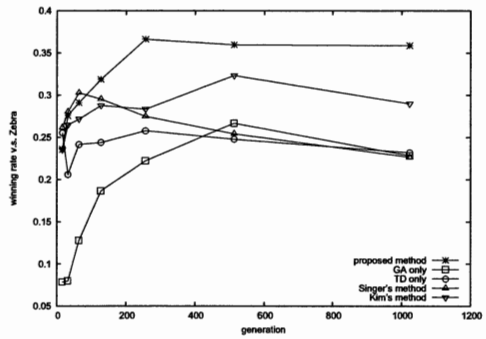

図 7 累計世代数を同一にした場合の勝率の推移

図 7 より, 各手法は途中から勝率が下降しており, 特定の世代数で止めて反復回数を増やした方が良い結果が得られることが分かる. しかし, 本手法は 256 世代以降もほとんど勝率が下がっておらず, 得られる解が反復回数を減らしても安定して求まっていることが読み取れる. TD only だけは 16, 32, 64 世代で勝率が大きくぶれてしまっているが, その原因は現在のところ分かっていない.

図 7 より, 各手法で最も勝率が高くなる世代 (最適停止世代) がおおまかに表 2 のように求まるため, この世代までの学習で得られた個体同士を直接に対戦させることで, 各手法で得られる個体の強さの比較を行った. 直接対決の結果は表 3 のようになった.

表 3 と図 6 の結果を比べると, 強さの順位が大きく異なることが分かる. 特に表 3 では GA only が大きく順位を上げており, また Singer's method と TD only が同程度の強さとなっている. しかしながら, いずれの場合においても本手法がもっとも好成績を上げていることは変わらず, 本手法の優位性が示される結果となった.

表 2 各手法と最適停止世代

手法	最適停止世代
proposed method	256
GA only	512
TD only	256
Singer's method	64
Kim's method	512

表 3 各手法間での勝率

自分＼相手	prop	GA	TD	Singer's	Kim's
proposed		58.0	69.4	69.5	54.3
GA only	42.0		56.7	57.4	46.6
TD only	30.6	43.3		49.6	29.4
Singer's	30.5	42.6	50.4		33.6
Kim's	45.7	53.4	70.6	66.4	

5. おわりに

本稿では, GA と TD(λ) 学習をハイブリッド GA の観点から組み合わる手法を提案し, その性能をオセロの局面評価パラメータの調整を例に評価した. 実験の結果, 同一リソースを用いた従来手法に対して 54.3%〜69.5%の勝率を記録し, ハイブリッド GA に基づいて GA と TD(λ) 学習を融合して用いることが非常に有用であるという結論が得られた.

今後の課題としては, 2.2.2 節で述べた SAHGA の本手法への適用が考えられる. 3 章で述べたように, 本手法は NAHGA というハイブリッド HGA の手法をベースに構成されている. NAHGA では各種パラメータを固定として扱うが, 例えば TD 確率を例にとると, 図 4 より前半の勝率の伸びは TD 確率が大きい方が, 後半の伸びは TD 確率が小さい方がよい結果が得られていることが分かる. SAHGA はこれらのパラメータを集団の状況によって変化させていく手法であるため, 前述の結果を考慮すれば性能が向上する可能性は高い. しかし一方で, SAHGA では絶対的な適応度が定義されている必要がある. ゲームの局面評価パラメータの場合, 適応度はその強さとなるが, ゲームプレイヤの強さを絶対的な数値として求めることは困難である. そのため, 本手法では GA の選択を個々の対戦の結果という相対的な強さを用いて行っていた. 絶対的な強さを定義する方法としては, ある一つのプ

レイヤー (例えば Zebra) に対する勝率として適応度を定義することが考えられる．しかし，この方法は基準として選んだプレイヤーによって結果が左右される，他のプレイヤーを用いることで手法の汎用性が犠牲になるといった欠点も含んでいる点に注意しなければならない．

また，本手法では TD(λ) 学習を GA と組み合わせているが，これを TDLeaf(λ)[17]，TDMC(λ)[18]，iLSTD(λ)[19] といった別の学習手法に置き換えてることでさらに強いプレイヤを得られる可能性があると考えられる．

謝辞 本研究の一部は文部科学省科学研究費補助金特定領域研究「情報爆発に対応する高度にスケーラブルなソフトウェア構成基盤」の助成を得て行われた．

参 考 文 献

1) R. S. Sutton and A. G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, 1998.
2) S. M. Lucas and T. P. Runarsson. Temporal Difference Learning Versus Co-Evolution for Acquiring Othello Position Evaluation. In *IEEE Symposium on Computational Intelligence and Games*, 2006.
3) 棟朝雅晴．遺伝的アルゴリズム　その理論と先端的手法．森北出版株式会社, 2008.
4) D. Whitley. A genetic algorithm tutorial. *Statistics and Computing*, Vol.4, No.2, pp. 65–85, 1994.
5) T. A. El-Mihoub, A. A. Hopgood, L. Nolle, and A. Battersby. Hybrid Genetic Algorithms: A Review. *Engineering Letters*, pp. 1816–0948, 2006.
6) K. E. Mathias, L. D. Whitley, C. Stork, and T. Kusuma. Staged hybrid genetic search for seismic data imaging. In *Evolutionary Computation, 1994. IEEE World Congress on Computational Intelligence., Proceedings of the First IEEE Conference on*, pp. 356–361, 1994.
7) F. P. Espinoza, B. S. Minsker, and D. E. Goldberg. A self adaptive hybrid genetic algorithm. In *Proceedings of the Genetic and Evolutionary Computation Conference 2001*, p. 759, 2001.
8) D. Whitley, S. Gordon, and K. Mathias. Lamarkian Evolution, the Baldwin effect, and function optimisation. In *Parallel Problem Solving from Nature–PPSN III: International Conference on Evolutionary Computation, the Third Conference on Parallel Problem Solving from Nature, Jerusalem, Israel, October 9-14, 1994: Proceedings*. Springer, 1994.
9) T. Elmihoub, A. A. Hopgood, L. Nolle, and A. Battersby. Performance of hybrid genetic algorithms incorporating local search. In *Proceedings of 18th European Simulation Multiconference*, pp. 154–160, 2004.
10) J. A. Singer. Co-evolving a Neural-Net Evaluation Function for Othello by Combining Genetic Algorithms with Reinforcement Learning. *LECTURE NOTES IN COMPUTER SCIENCE*, pp. 377–389, 2001.
11) K. J. Kim, H. Choi, and S. B. Cho. Hybrid of Evolution and Reinforcement Learning for Othello Players. In *Computational Intelligence and Games, 2007. CIG 2007. IEEE Symposium on*, pp. 203–209, 2007.
12) CEC 2006 Othello Competition Results. http://algoval.essex.ac.uk:8080/othello/html/cec2006results.html.
13) InTrigger's homepage. http://www.intrigger.jp.
14) M. Buro. An Evaluation Function for Othello Based on Statistics. *NEC Research Institute*, 1995.
15) LOGISTELLO's homepage. http://www.cs.ualberta.ca/mburo/log.html.
16) Zebra's homepage. http://www.radagast.se/othello/.
17) J. Baxter, A. Tridgell, and L. Weaver. Learning to Play Chess Using Temporal Differences. *Machine Learning*, Vol.40, No.3, pp. 243–263, 2000.
18) 大崎泰寛，柴原一友，但馬康宏，小谷善行. TDMC(λ) に基づく評価関数の調整．第 13 回ゲームプログラミングワークショップ, pp. 73–79, 2008.
19) A. Geramifard, M. Bowling, M. Zinkevich, R. S. Sutton, and A. Edmonton. iLSTD: Eligibility Traces and Convergence Analysis. In *Advances in Neural Information Processing Systems: Proceedings of the 2006 Conference*. MIT Press, 2007.