2008－EC－9 (2)
2008／3／7

戦略型カードゲームのための戦略獲得法

藤井叙人[†] 橋田光代[†,†††] 片寄晴弘[††,†††]

市販テレビゲームにおいては，CPUの人間らしさというリアリティに，プレイヤの意識が高まりつつある．従来研究では，将棋や仮想空間におけるCPUの人間らしさの検討はなされているものの，市販テレビゲームのような戦略型ゲームにおいて，人間らしい行動や戦略をAI技術を用いて実現した例はほとんどない．本稿では，CPUの「人間らしさ」の実現を目標とし，戦略型カードゲームの戦略を自動的に獲得する機構を提案した．戦略学習における困難性として，部分観測に起因した巨大な状態空間が挙げられるが，サンプリング，相手の行動予測，ゲームの特徴を考慮した次元圧縮により克服した．また，戦略学習機構によって得た戦略を，ルールベースの戦略と対戦させ，有効性を評価した．

Strategy-acquisition Scheme for Strategic Card Game

Nobuto Fujii,[†] Mitsuyo Hashida[†,†††]
and Haruhiro Katayose[††,†††]

The computer(CPU) like a human has lately attracted considerable attention in the video game. The past studies report strategy-acquisition scheme for the board game such as shogi. However, there are few studies about these scheme applied in the video game. We aim at the acquisition of strategy like a human, we present an automatic strategy-acquisition scheme for strategic card game. Because of this card game includes many unobservable variables in a large state space, we suggest a sampling technique, action predictor, and value function. To evaluate our method, we carried out computer simulations where our agent played against a rule-based agent.

1. はじめに

市販テレビゲームやPC用ゲームにおいては，これまで，プレイヤのゲームに対する面白さの向上を目的として，数多くの工夫がなされてきた．しかし，ハードウェアの発展や，インターネット上でのエンタテインメントの普及に伴って，プレイヤのゲームに対するリアリティの要求は増す一方である．グラフィックやサウンドなど，ゲームの各要素におけるリアリティだけではなく，ゲーム内のコンピュータ (CPU) の**人間らしさ**，というリアリティに対してプレイヤの意識が高まりつつあり，近年，CPUの**人間らしさ**を追求した研究が盛んに取り組まれている[1)~9)]．

CPUの人間らしさを感情に伴う行動の表出という視点から検討した研究としては，キャラクタの感情表現の有効性の評価[1),2)]や，プレイヤの操作に応じてキャラクタが自律的な行動を始めるインタラクティブゲーム[3)]などがある．また，ゲームにおけるCPUの戦略に着目してCPUの人間らしさを追求した研究として，将棋[4)]やチェス[5)]などのボードゲームや，トランプゲーム[6)~9)]への戦略学習機構の実装が挙げられる．これらの研究では，実際の対戦データからCPUの戦略を学習させ，相手の戦略に対して臨機応変に人間らしい行動を出力するような戦略学習機構の実現が目的とされている．コンピュータ将棋では，学習により戦略を得たCPU同士の世界大会や，CPU対プロ棋士の対局など，研究を促進するための活動が活発に行われているほどである[10)]．

将棋やトランプゲームと違い，市販テレビゲームやPC用ゲームはいくつかのジャンルに分けられるものの，個々の仕様はゲームによって異なる．そのため，

† 関西学院大学大学院理工学研究科
Graduate School of Science and Technology, Kwansei Gakuin University
email: nobuto@ksc.kwansei.ac.jp
†† 関西学院大学理工学部
School of Science and Technology, Kwansei Gakuin University
††† 科学技術振興機構 戦略的創造研究推進事業 CrestMuse プロジェクト
CrestMuse Project, CREST, JST

CPUの戦略や振る舞いなどの作り込みには，ゲームタイトルごとに違った実装方法を強いられているのが現状である．ルールベースやパラメータによる疑似的な人間らしさの作り込みはなされているものの，学習機構によるCPUの戦略獲得や行動制御はほとんど実現されておらず，CPUの人間らしさというリアリティの検討はなされていない．

本稿では，市販テレビゲームにあるような戦略型カードゲームを題材として，自動的に戦略を学習する汎用的な戦略学習機構を提案し，課題について議論する．CPUの戦略学習には文献6)~9)で提案されている手法を応用する．以下，第2章で関連研究を紹介し，第3章で問題の設定と定式化，第4章で計算機実験による戦略学習機構の評価，第5章で戦略学習機構の汎用性を検討し，最後に考察を行う．

2. 関連研究

CPUの人間らしさを感情に伴う行動の表出という視点から検討した研究として，中村らが，仮想空間内のCPUにおける人間らしい感情表現の有効性を実証している[2]．これは，小林らが提案したモーションオーバーラッピング[11]を，仮想空間内のCPUへ適用したものである．ドアの前で左右に動くというマインド表出を行っているCPUに対し，それを観察したプレイヤがドアを開けるかどうか，という検証の結果，CPUの意思をプレイヤにさりげなく伝え，プレイヤに能動的な行動を起こさせるよう誘導できることが示されている．

また，中野らは，プレイヤが仮想空間内にオブジェクトを加えたり，CPUに手で触れることによって，CPUが人間らしい感情を表出し，自律的な行動を開始するインタラクティブゲームを提案している[3]．CPUの行動は，エピソード(ストーリー)の流れを管理するツリー，挨拶や台詞などのリアクションを管理するツリー，キャラクタの移動などのスケジュールを管理するツリー，といった複数のツリー構造により制御されている．人間らしい感情に基づいた動作をするキャラクタの実現は，プレイヤと相互作用することで，ゲームの新たな楽しみ方を提供できると考察している．

ゲームにおけるCPUの戦略に着目してCPUの人間らしさを追求した研究として，藤田らは，トランプゲームの“Hearts”を対象としたCPUの戦略学習の手法を提案している[6)~9)]．藤田らは，トランプを52枚用いるため巨大な状態空間となること，相手の所持するカードは観測できないため部分観測状況となること，4人対戦のマルチエージェントゲームであることの3つを，Heartsにおける戦略学習の困難性と考察している．その上で，困難性の解決手法として，パーティクルフィルタによるサンプリング，相手の行動を予測する行動予測器，現在の状態を評価する状態価値関数，ゲームの特徴に基づく次元圧縮を提案している．計算機実験として，提案手法に基づく学習エージェントと，ルールベースエージェント3体とを対戦させた結果，約2,000ゲーム学習後にはルールベースエージェントよりも強い戦略を獲得している．さらに，約4,000ゲーム学習後には，人間の熟達者よりも優れた戦略を得ることに成功している．藤田らは，提案手法による戦略学習がHeartsだけではなく，他の部分観測カードゲームへも応用が可能であり，問題に依存した様々な状況に対しても適用することができると検討している．

CPUの人間らしさのリアリティに着目した研究は以上に挙げたものなどがあるが，市販テレビゲームにあるような戦略型ゲームを取り上げ，かつ，人間らしい行動や戦略をAI技術を用いて実現した研究はほとんど報告されていない．

3. 問題の設定と定式化

本稿における戦略型カードゲームのルールと，戦略的要素である属性関係について述べる．また，戦略学習機構を実装する上での困難性を考察し，問題解決のための手法を提案する．

3.1 ゲームのルール

本稿で用いる戦略型カードゲームは，プレイヤ対CPUの対戦型ゲームとし，ルールは以下のように設定する．

(1) プレイヤ，CPUは10体のモンスターの中から，3体選択する．

(2) 選択した3体から，戦闘状態とするモンスター1体を選択する．残り2体を待機状態とする．(互いに，戦闘状態のモンスターしか観測できない．)

(3) 戦闘状態のモンスターに対して，攻撃，特殊攻撃，入れ替えを指示する．入れ替えとは，現在戦闘状態のモンスターを待機状態とし，代わりに待機状態のモンスターの中から戦闘状態にするモンスター1体を選択する．

(4) プレイヤ，CPUの戦闘状態のモンスターが，相手の戦闘状態のモンスターに対して，指示された行動を行う．攻撃，特殊攻撃の場合は相手モンスターにダメージを与えることになる．

(5) 攻撃され，モンスターの体力が0になれば，そ

攻撃する側＼攻撃される側	火	水	雷	地	ノ
火	△	△	―	○	―
水	○	△	―	○	―
雷	―	○	△	×	―
地	○	―	○	―	―
ノ	―	―	―	―	―

ダメージ倍率
○： 2倍
―： 1倍
△：1/2倍
×： 0倍

表 1 モンスターの属性表．ノはノーマルを表す．
Table 1 Table of Elements

のモンスターは戦闘不能とする．

(6) どちらかのモンスターが 3 体とも戦闘不能になるまで，4，5 を繰り返す (以下，4，5 をまとめて 1 ステップと呼ぶ)．

戦略的な要素を付け加えるため，各モンスターには属性を設定する (**表 1**)．属性により，モンスターごとに，得意なモンスター，苦手なモンスターが決定することになる．つまり，現在戦闘状態の相手モンスターに対して有利に戦えるモンスターを，手持ちのモンスター 3 体の中から選択する必要がある．

3.2 戦略学習法の検討

3.1 節で述べた戦略型カードゲームを，戦略学習対象の観点から整理すると以下のようになる．

巨大な状態空間をもつ プレイヤと CPU が持っているモンスターの組み合わせを**状態**と定義する．両者が 10 体のモンスターから 3 体を選択した状態数に加え，体力が減った状態や，モンスターが戦闘不能である状態も考慮しなくてはならない．

部分観測となる 相手が所持するモンスターは，戦闘状態となるまで観測することができない．つまり，相手の所持するモンスターを推定する必要があり，状態空間はさらに巨大になる．

教師が存在しない 戦略型ゲームにおいて，絶対的に正しい戦略は存在せず，状況に応じた戦略を獲得する手法が必要となる．

ゲームにおける価値に遅れが生じる ゲームを有利に進めるための価値を，相手に与えるダメージと自分が受けるダメージから求める．各ステップで自分のモンスターが相手モンスターに攻撃をする際に得られる価値の他に，相手モンスターを倒したときの価値や，ゲームに勝利したときの価値など，ゲーム全体を通して見た時の将来的な価値 (遅れ価値) が重要な役割を占める．

戦略を学習する上で有効な学習法はいくつか提案されており，研究例も少なくない．例えば，隠れマルコフモデル (HMM) による格闘技の行動パターンの学習[12]，リカレント型ニューラルネットワーク (RNN) による聴覚情報と視覚情報の対応関係の学習[13]，ベイジアンネットワークによるエージェントの表情表出の学習[14] などが挙げられる．しかし，巨大な状態空間による計算量の増大，教師なしであること，遅れ価値が重要であることを考慮に入れると，本稿の戦略型カードゲームにおける学習法として適しているとはいえない．

そこで，本稿では強化学習法[15] を用いる．学習データから各状態における状態価値を学習するため，遅れ価値が十分に反映される．また，部分観測に起因する巨大な状態空間は，パーティクルフィルタ[16] 等のサンプリング法により解決できる．強化学習法は，ゲーム中に得られる報酬から，試行錯誤を通じて戦略を学習するため，人間のエキスパートより優れた戦略を得られる可能性が期待されている学習法でもある[6)~9)]．

3.3 学習機構の実装

戦略学習機構は，各ステップでの最適行動選択を目的とし，以下の 6 部分から構成される (**図 1**)．

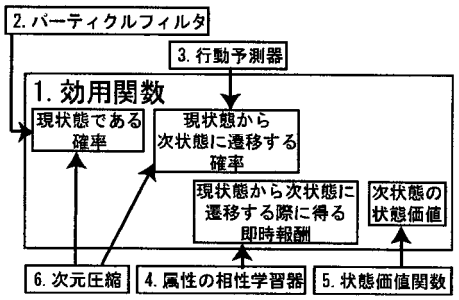

図 1 戦略学習機構
Fig. 1 Strategy-acquisition System

1. 各ステップでの最適行動を決定する効用関数
効用関数を最大とする行動が最適行動であると定義する．さらに，効用関数を “現状態である確率 × 現状態から次状態に遷移する確率 × 現状態から次状態に遷移する際に得る即時報酬 × 次状態の状態価値” と定義する．しかし，巨大な状態空間をもち，かつ，部分観測ゲームであるために，現状態の候補と，そこから遷移する次状態の候補が大量に存在し，そのすべてにおいて効用関数を計算するのは不可能である．そこで，パーティクルフィルタによりサンプリングし，近似的に現状態と次状態を推定する．

2. パーティクルフィルタによるサンプリング
前状態で得たサンプルに対して，既知である前状

態のプレイヤと CPU の行動を適用することで，各サンプルは現状態の候補として扱うことが可能になる．この作業はゲームのルールに矛盾しないように行うことで，得られた現状態のサンプルは“現状態である確率”の分布と一致する．

3. 相手の行動を予測する行動予測器

“現状態から次状態に遷移する確率”は，相手の行動を予測する行動予測器により近似的に推定する．現状態において，プレイヤの行動と CPU の行動が決定すれば，次状態は一意に決まる．つまり，行動予測器により求めたプレイヤの行動選択確率が，“現状態から次状態に遷移する確率”と一致する．行動予測器の学習には，ニューラルネットのひとつである多層パーセプトロン (MLP) を用い，入力は現状態の真の観測，教師は実際にプレイヤがとった行動とする．

4. 属性の相性の学習器

“現状態から次状態に遷移する際に得る即時報酬”の一つに，属性の相性を用いる．属性の相性は，本稿の戦略型カードゲームにおいて最も重要な要素であり，次状態に遷移した際の属性の相性は，ゲームの勝敗を大きく左右する鍵となる．即時報酬は，属性の相性の他に，相手に与えたダメージの量，自分が受けたダメージの量，相手のモンスターを倒した時，ゲームに勝利した時にも与えられる．属性の学習にも MLP を用い，入力はプレイヤと CPU のモンスターの属性，教師はその時に与えることができたダメージの量とする．

5. 状態の価値を推定する状態価値関数

“次状態の状態価値”は，状態の価値を推定する状態価値関数により求める．状態価値関数にも MLP を用い，1 ゲームが終わる毎にその履歴から学習する．入力は真の現状態，教師はその時刻からゲーム終了までの即時報酬と属性の相性との和とする．

6. ゲームの特徴による次元圧縮

戦略型カードゲームのゲーム状態の次元は，10 体のモンスターに加えて，モンスターの属性，各モンスターの体力や攻撃力などのパラメータ，モンスターが戦闘不能かどうか，などすべてを考慮すると非常に高い次元となる．そこで，上記 1〜5 における，“状態”，“観測”，“行動”にはゲームの特徴による次元圧縮が施されている．次元圧縮によって無駄な情報を省くことで，戦略学習機構は効率よく学習することが可能となる．

以下に “状態”，“観測”，“行動”の圧縮方法を示す．

状態 (32 次元)

0〜17 次元は，CPU とプレイヤの戦闘状態モンスターに関する情報．

0-4 : CPU の属性が，火，水，雷，地，ノのどれか？
5-9 : プレイヤの属性が，火，水，雷，地，ノのどれか?
10，11 : CPU，プレイヤの体力
12，13 : CPU，プレイヤの素早さ
14 : CPU の攻撃力−プレイヤの防御力
15 : プレイヤの攻撃力− CPU の防御力
16 : CPU の特殊攻撃力−プレイヤの特殊防御力
17 : プレイヤの特殊攻撃力− CPU の特殊防御力
18-22 : CPU が所持する火，水，雷，地，ノ属性のモンスターの数
23-27 : プレイヤが所持する火，水，雷，地，ノ属性のモンスターの数
28，29 : CPU，プレイヤの死んでいるモンスター数
30，31 : CPU，プレイヤの生きているモンスターのＨＰの合計

観測 (25 次元)

“状態”の 32 次元から

18-22 : CPU が所持する火，水，雷，地，ノ属性のモンスターの数
30，31 : CPU，プレイヤの生きているモンスターのＨＰの合計

を取り除いたもの．

行動 (7 次元)

0 : 攻撃
1 : 特殊攻撃
2 : 火属性モンスターがいれば入れ替え
3 : 水属性モンスターがいれば入れ替え
4 : 雷属性モンスターがいれば入れ替え
5 : 地属性モンスターがいれば入れ替え
6 : ノ属性モンスターがいれば入れ替え

“状態”，“観測”の次元圧縮では，重要な特徴に対して多くの次元を割り当てることで，より効率の良い学習が可能である．本稿の戦略型カードゲームの場合は，戦闘状態モンスターの属性と，所持しているモンスターの属性が極めて重要であるため，CPU とプレイヤにおいてそれぞれ 5 次元ずつ割り当てている．また，“状態”の 10〜17 次元目と 30〜31 次元目の値に関しては対数をとっている．他の次元がブール変数や一桁の実数であるの

に対し，2桁から4桁の実数となるため，MLPの学習が効率よく進まない可能性があるからである．

本稿における戦略学習機構は1を最大の目的とし，それに付随する形で3〜5を学習することが目標であるといえる．

4. 計算機実験

本稿における戦略学習機構に基づく学習エージェント (RL-agent) を，ルールベースエージェント (Rule-based) と対戦させることで，戦略学習機構の有効性を評価する．

RL-agentが100ゲーム学習する毎に，評価ゲームとしてRule-basedと200ゲーム，ランダムに行動選択を行うエージェント (Random) と200ゲーム対戦し，RL-agentの勝率などを求めた．本稿の戦略型カードゲームの勝敗は，モンスター3体の組み合わせに大きく依存するため，あらかじめ200ゲーム分のモンスターの組み合わせをランダムに決定しておき，評価ゲームでは毎回そのモンスターの組み合わせを用いている．一方，学習ゲームにおけるモンスターの組み合わせは完全にランダムで決定した．

Rule-basedの個性として，攻守のバランスがとれた“バランス型”，攻撃志向の“力押し型”，防御志向の“堅実型”を用意した．

4.1 Rule-basedのルール

実験に用いたRule-basedは以下の9つのルールを持ち，(1)を最優先ルールとして，以下順番に優先度が低くなるよう設定されている．

(1) 自分の戦闘状態モンスターの属性 (自属性) が，相手の戦闘状態モンスターの属性 (敵属性) に対し「○」ならば特殊攻撃．

(2) 自属性が敵属性に対し「○」となるモンスターに入れ替え．

敵属性が自属性に対し「○」ならば，

(3) 「×」となるモンスターに入れ替え．

(4) 「△」となるモンスターに入れ替え．

(5) 「−」となるモンスターに入れ替え．

自属性が敵属性に対し「×」ならば，

(6) 「−」となるモンスターに入れ替え．

(7) 「△」となるモンスターに入れ替え．

(8) 自属性が敵属性に対し「△」ならば，「−」となるモンスターに入れ替え．

(9) 攻撃か特殊攻撃か，ダメージの大きい方を選択．

このルールは攻守のバランスがとれた“バランス型”Rule-basedである．個性を変えたものとして，攻撃志向の“力押し型”Rule-basedはルール6〜8をルール3よりも優先度が高くなるように設定してあり，できる限り相手に多くのダメージを与えられるような行動を選択する．また，“堅実型”Rule-basedはルール3〜5をルール2よりも優先度が高くなるように設定してあり，できる限り自分がダメージを受けないような行動を選択する．

4.2 結果と考察

実験結果を**図2〜図6**に示す．

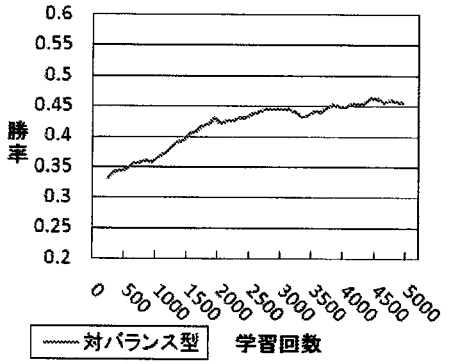

図2 RL-agentの勝率 (対バランス型)
Fig. 2 RL-agent's winning percentage(vs Balance-Type)

図2は，RL-agentとバランス型Rule-basedを5,200ゲーム対戦させた際のRL-agentの勝率である．横軸は学習したゲーム数，縦軸はRL-agentの勝率を表す．グラフは，3回の学習過程における500ゲーム間の移動平均である．RL-agentの勝率は，学習をしていない段階では3割程度であったが，学習開始から徐々に増加し，5,200ゲーム学習後では5割弱となることが分かる．5,200ゲーム以上の学習も試みたが，戦略は振動し収束せず，5割弱以上の勝率を得られなかった．

図3はRL-agentと力押し型Rule-basedを，**図4**はRL-agentと堅実型Rule-basedを5,200ゲーム対戦させた際のRL-agentの勝率である．横軸，縦軸，グラフに関しては図2と同様である．RL-agentの勝率は，学習していない段階では3割程度，5,200ゲーム学習後では5割弱であり，図2と同じような結果になった．5,200ゲーム以上の学習も試みたが，戦略は振動し収束せず，5割弱以上の勝率を得られなかった．

RL-agentと3つのRule-basedとの実験結果 (図2，図3，図4) において，RL-agentの勝率が5割弱であることから，各Rule-basedの戦略を真似る程度の行動

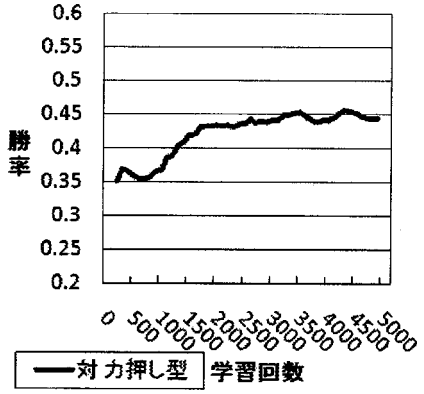

図 3 RL-agent の勝率 (対 力押し型)
Fig. 3 RL-agent's winning percentage(vs Offensive-Type)

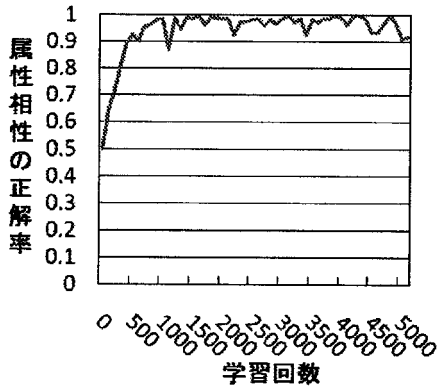

図 5 属性学習の正解率
Fig. 5 correct rate of Elements

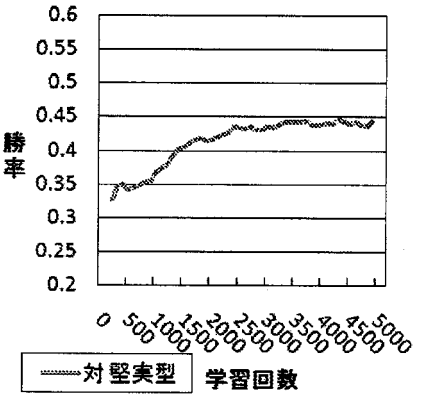

図 4 RL-agent の勝率 (対 堅実型)
Fig. 4 RL-agent's winning percentage(vs Defensive-Type)

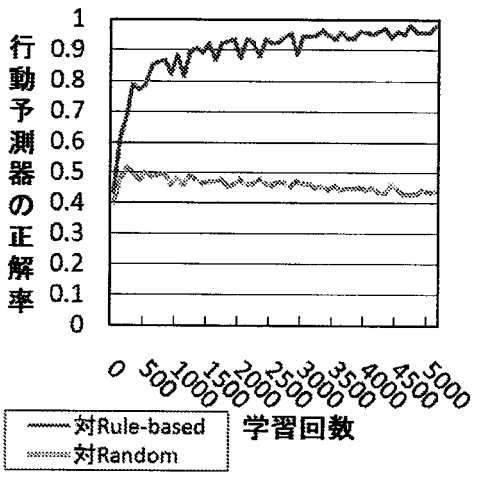

図 6 行動予測器の正解率
Fig. 6 Action Predictor's correct rate

選択が可能となっていることが分かる．しかし，Rule-based よりも強い戦略を得るには至らない．また，どの Rule-based と対戦した場合も，RL-agent の勝率が 5 割弱まで上昇したことから，Rule-based の個性が変化しても，対戦相手の戦略に応じて RL-agent は戦略学習ができているといえる．この結果により，本稿の戦略学習機構には汎用性があることが示された．

図 5，図 6 は，RL-agent が 5,200 ゲームの学習をした際の，属性の強弱の正解率と行動予測器の正解率である．横軸は学習したゲーム数，縦軸はそれぞれ，属性の強弱の正解率，行動予測器の正解率を表す．グラフは，3 回の学習過程に対する正解率の平均である．属性の強弱の正解率とは，RL-agent が，自分の属性と相手エージェントの属性の相性を正確に学習できているかどうかを示す．学習開始から急激に増加し，1,000 ゲーム学習後にはほぼ正確に属性の強弱を判断できていることが分かる．また，行動予測器の正解率とは，行動予測器の出力と相手エージェントの行動が一致したかどうかを示す．学習をしていない段階では 4 割程度だが，対バランス型 Rule-based においては，学習開始から急激に増加し，1,500 ゲーム学習後には 9 割程度の行動予測が可能となっている．それ以降も学習することで徐々に増加し，5,200 ゲーム学習後にはほぼ正確に相手の行動を予測している．しかし，対 Random においては行動予測器の正解率はほとんど変化しない．

属性の強弱の正解率 (図 5)，行動予測器の正解率 (図 6) から，対 Rule-based において，相手エージェントの行動予測と属性の強弱が正確に学習できていることは確認できる．つまり，RL-agent と Rule-based

の実験結果 (図 2〜図 4) において，RL-agent の勝率が Rule-based を上回らない原因として，状態価値関数の学習が振動し収束していないと思われる．節 3.3 で述べた，状態価値関数を学習する際の教師が，ゲームの状態価値を適切に表しているか検討し，学習が進むにつれて学習パラメータの振れ幅を小さくするような手法が必要である．また，もう一つの原因として，本稿で設定した戦略型カードゲームのルールにおいては，Rule-based の戦略がほぼ最適である可能性も考えられる．人間が作り込んだルールベースの戦略では最適解が見つからないような，戦略性の高いゲームのルールを設定する必要があるかもしれない．

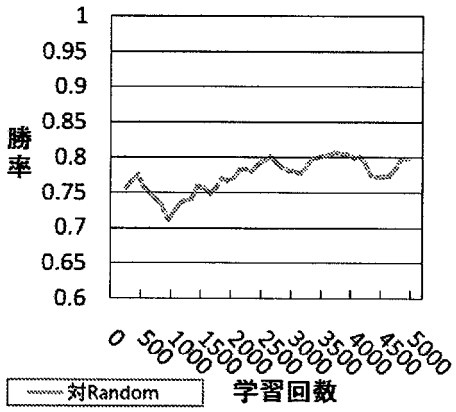

図 7 RL-agent の勝率 (対 Random)
Fig. 7 RL-agent's winning percentage(vs Random)

図 7 は，RL-agent と Random を 5,200 ゲーム対戦させた際の RL-agent の勝率である．横軸は学習したゲーム数，縦軸は RL-agent の勝率であるが，図 2〜図 4 とは縦軸の範囲が異なる点に注意する．グラフは，3 回の学習過程に対する 500 ゲーム間の移動平均である．RL-agent の勝率は，学習をしていない段階では 7 割強であるが，5,200 ゲーム学習後でも 8 割程度にとどまり，それ以上の勝率は得られなかった．

図 8 は，モンスター入れ替えをしない RL-agent と Random を 5,200 ゲーム対戦させた際の RL-agent の勝率である．横軸，縦軸，グラフは図 7 と同様である．RL-agent の勝率は，学習をしていない段階では 8 割程度，1,000 ゲーム学習後では 8 割強の勝率を得たが，それ以降に勝率が上昇することはなかった．

RL-agent と Random との実験結果 (図 7) から，対 Random においては，対 Rule-based ほど勝率が上昇しないことが分かる．その原因は，相手の行動を予測する行動予測器の正解率が上昇しない (図 6) からである．Random は法則性のない行動選択であるため，行動が予測できないのは当然であり，RL-agent が Rule-based と対戦することで学習した戦略が，Random に対しては逆に不利になっていると考えられる．その証拠となるのが，対 Random において RL-agent のモンスター入れ替えを禁じた際の結果である (図 8)．法則性のない行動選択をする Random に対し，行動予測が外れた時のモンスター入れ替えのリスクがなくなるため，RL-agent の勝率はモンスター入れ替えを行う時よりも 1 割ほど高い．つまり，行動予測が外れた時のモンスター入れ替えは，極めてリスクが大きいことがわかる．

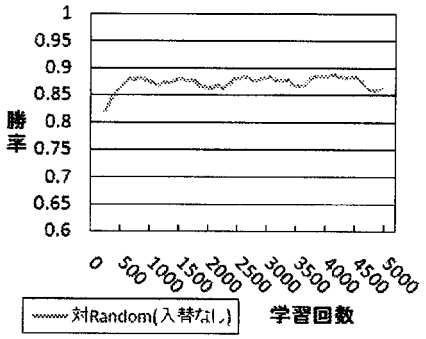

図 8 入れ替えなしでの対 Random
Fig. 8 vs Random(not change)

5. 検　　討

本稿で提案する戦略学習機構は，巨大な状態空間を持つ戦略型カードゲームにおいて，サンプリング，最適行動を決定する効用関数，属性の強弱の学習，相手の行動予測器，状態価値関数，ゲームの特徴による次元圧縮を用いることで，対戦相手の戦略から，自らの戦略の学習が可能であることが示された．

本手法では，戦略型ゲームを，確率的なゲームとして扱うことにより，近似的に戦略を得ている．そのため，本稿における戦略型カードゲームだけではなく，市販テレビゲームにあるような戦略型ゲームへも応用が可能である．例えば，ロールプレイングゲームや格闘ゲームなど，キャラクタや必殺技などに相性が設定されているゲームや，特定の行動がある対象に対して非常に有利である場合などは，本稿の属性表を用いることで有利不利を学習することができる．また，シミュレーションゲームなど，ある時系列でプレイヤと CPU が交互に行動をするゲームでは，真の状態を正確に推定し，相手の行動を予測することが極めて重要であるが，本稿の相手エージェントに対応する行動予

測器や，相手の行動予測に基づく真の状態の推定により実現できる．本手法は，相手の戦略の変化に適応でき，また，ゲームの難易度に応じた戦略なども学習により得ることが可能で，ゲーム性に依存した様々な状況に対しても適用可能と考えられる．

本手法は，各ステップでの最適行動選択を目的とした単層の学習機構である．しかし，実際にプレイヤがゲームを行う場合，プレイヤ自身に感情や個性があることに加えて，必勝である初期カードの選択を考えたり，対戦相手の個性を推察したりすると考えられる．より人間らしいCPUの戦略を獲得するためには，それぞれの問題において個別の学習機構をもつような，階層型戦略学習機構の検討が課題として残されている．

6. ま と め

本稿では，市販テレビゲームのような戦略型カードゲームを題材とし，人間らしい戦略の学習機構を提案した．

本手法により，ルールベースやパラメータによって作り込まれた，ゲーム性に依存する戦略ではなく，自動的にCPUの戦略を学習する汎用的な戦略学習が可能となった．巨大な状態空間による計算の困難性を克服し，相手の行動予測の結果から真の状態を正確に推定する必要があり，また，属性の強弱を学習することで現状態の状態価値を求め，相手の行動予測と現状態の状態価値から自身の最適行動を決定することが重要であった．本手法では，パーティクルフィルタによるサンプリングによって巨大な状態空間の計算困難性を克服した．ニューラルネットを用いることで相手の行動予測器と属性の強弱に依存した状態価値を求め，真の状態を推定し，最適と思われる行動を決定した．状態空間の次元が高く，無駄な情報が多く含まれる問題は，ゲームの特徴を考慮した上で次元圧縮を行うことで解決した．本稿で提案した戦略獲得機構により得た戦略を，9つのルールをもつルールベースエージェントの戦略と対戦させることで，戦略学習機構の有効性を評価した．

今後は，効果的に学習を進める手法の検討を行い，ルールベースよりも良い戦略の獲得を目指す．また，対戦相手の性格の推察や，感情に基づく戦略の変化など，より戦略的な要素を考慮することが可能な戦略学習機構を検討する予定である．

参 考 文 献

1) 大野陽介, 鴨崎真直, ラック・ターウォンマット: “物語生成システムにおける感情を持ったNPCの動作の適切さの検証”, 情報処理学会研究報告, Vol. 2006, No. 134, 2006-EC-5, pp. 25-30 (2006)
2) 中村知貴, 澤淳二, ラック・ターウォンマット: “仮想空間におけるモーションオーバーラッピングの有効性の検証”, 情報処理学会研究報告, Vol. 2006, No. 134, 2006-EC-5, pp. 19-24 (2006)
3) 中野敦, 河村仁, 三浦枝里子, 星野准一: “Spilant World:エピソードツリーによるインタラクティブなストーリー創発型ゲーム”, 芸術科学会論文誌, Vol. 6, No. 3, pp. 145-153 (2007)
4) 保木邦仁: “コンピュータ将棋における全幅探索と futility pruning の応用”, 情報処理学会学会誌, Vol. 47, No. 8, pp. 884-889 (2006)
5) Feng-Hsiung Hsu: “IBM's Deep Blue Chess grandmaster chips”, Micro IEEE, Vol. 19, pp. 70-81 (1999)
6) 藤田肇, 石井信: “マルチエージェントカードゲームのための強化学習法の改良”, 電子情報通信学会技術研究報告, Vol. 102, No. 731, pp. 167-172 (2003)
7) S. Ishii, H. Fujita: “A Reinforcement Learning Scheme for a Partially-Observable Multi-Agent Game”, Machine Learning, Vol. 59, pp. 31-54 (2005)
8) 藤田肇, 石井信: “部分観測カードゲームのためのモデル同定型強化学習”, 電子情報通信学会論文誌, Vol. J88-D-II, No. 11, pp. 2277-2287 (2005)
9) H. Fujita , S. Ishii: “Model-Based Reinforcement Learning for Partially Observable Games with Sampling-Based State Estimation”, Neural Computation, Vol. 19, pp. 3051-3087 (2007)
10) 伊藤毅志: “世界コンピュータ将棋選手権報告”, 情報処理学会学会誌, Vol. 48, No. 7, pp. 775-779 (2007)
11) 小林一樹, 山田誠二: “擬人化したモーションによるロボットのマインド表出”, 人工知能学会論文誌, Vol. 21, No. 4, pp. 380-387 (2006)
12) 高野渉, 山根克, 中村仁彦: “運動の認識・生成に基づく原始的コミュニケーションの階層構造モデル”, 日本ロボット学会学術講演会予稿集 (2005)
13) 尾形哲也, 小嶋秀樹, 駒谷和範, 奥野博: “RNNPB による視聴覚情報変換を利用したロボットの身体・音声表現” 電子情報通信学会技術研究報告, TL-2006-22, NLC-2006-18, PRMU2006-99, pp. 45-50 (2006)
14) 湯浅将英, 安村禎明, 新田克己: “オンライン交渉における擬人化エージェントの表情選択支援”, ヒューマンインタフェース研究会報告, Vol. 2004, No. 74, 2004-HI-109, pp. 1-6 (2004)
15) R.S. Sutton and A.G. Barto: “Reinforcement Learning”, An Introduction, MIT Press, (1998)
16) 北川源四郎: “モンテカルロ・フィルタおよび平滑化について”, 統計数理, Vol. 44, No. 1, pp. 31-48 (1996)