トランプゲームに対するSVRを適用したFALCONの改良

Improvement of FALCON using SVR for a card game

笠原　和真[1*]　伊藤　崇[1]　高橋　健一[1]　稲葉　通将[1]
Kazuma KASAHARA[1]　Takashi ITO[1]　Kenichi TAKAHASHI[1]　Michimasa INABA[1]

[1] 広島市立大学大学院
[1] Graduate School of Information Sciences, Hiroshima City University

Abstract: This paper proposes a method to improve the learning performance of a learning agent with FALCON, to make a player agent for a card game “hearts”, which is one of the multi-player imperfect-information games. FALCON is a machine learning method which is an extended fuzzy ART(Adaptive Resonance Theory). The previous work showed that FALCON is effective for hearts. In this study, to improve the learning performance, the action set of the agent is changed based on strategies of hearts, and a method that employs a prediction by the support vector regression is proposed.

1　はじめに

近年，人工知能を用いた研究が多くなされ，その技術はゲームやロボットなどの分野で広く用いられている [1]. 最近では，人工知能の技術をゲームに応用した研究の一つとして，囲碁やチェスといった二人完全情報ゲームが注目されている．二人完全情報ゲームとは，プレイヤが 2 人で，局面情報が全てのプレイヤに与えられているゲームである．これに対して，3 人以上のプレイヤが存在し，局面に不完全な情報があるゲームを多人数不完全情報ゲームという．一般的に，不完全情報ゲームは完全情報ゲームよりも考慮すべき要素が多く，また確率的な要素も多分に含まれている．したがって，多人数不完全情報ゲームは不確定な情報を多く含む実世界に近い問題といえる．このような環境に対して効率的に強化学習を適用するためには，知覚状態空間を適切に離散化する必要がある．状態空間を離散化する手法は，タイルコーディング，ニューラルネットワーク，ファジィ推論などのオフライン手法，及び ART(Adaptive Resonance Theory) などのオンライン手法に大別することができる．

本研究では，ART を拡張した機械学習手法であり，二人完全情報ゲーム及び多人数不完全情報ゲームに対して有効性が示されている FALCON(a Fusion Architecture for Learning, COgnition, and Navigation)[2] を使用し学習実験を行う．FALCON は Ah-Hwee Tan が提案したオンライン学習手法であり，知覚，行動，報酬全てのベクトルに対して同時に複数のマッピングを学習することにより，知覚情報空間の離散化及び行動規則の学習を同時に実施することができる．二人完全情報ゲーム及び多人数不完全情報ゲームに対して FALCON を用いる研究がなされ，有効性が示されている [3][4]．この FALCON を多人数不完全情報ゲームであるカードゲームのハーツに適用し，FALCON の性能向上を図る．我々はこれまでの研究において，ハーツに対する FALCON の改良手法を提案し，より強い学習エージェントを構築することに成功した [5]．また，FALCON を用いて訓練した学習器を複数個組み合わせることで，より強いエージェントを構築する手法を提案した [6]．しかし，モンテカルロシミュレーションにより行動を選択するエージェントに勝つことはできなかった．そこで本研究では，FALCON の改良手法を提案し，より強い学習エージェントを構築することを目的とする．まず，エージェントをより戦略的に行動させるため，FALCON の行動種類を実際のハーツにおける戦略に基づいて再設定した．次に，効率的な学習を行うため，FALCON の学習時における行動選択に対してサポートベクター回帰による予測を適用した．また，勝率を上げるため，ゲーム局面ごとに複数の FALCON を使用し，学習進行度に応じた相手エージェントの変更，学習フィードバックの変更を行った．これらの改良を適用した学習エージェントの性能比較を行う．

*連絡先：広島市立大学大学院情報科学部知能工学専攻
〒 731-3194　広島県広島市安佐南区大塚東三丁目 4 番 1 号
E-mail: kasahara@cm.info.hiroshima-cu.ac.jp

2 ハーツ

2.1 ルール

トランプゲームのハーツは一般的に 4 人でプレイされ，ジョーカーを除いた 52 枚のカードを用いる．A, K, Q, ..., 4, 3, 2 の順で強さの順位付けがされており，スート間の優劣はない．各プレイヤはじめに 13 枚のカードを手札として持つ．親から順に全員が 1 枚ずつカードを出すことをトリックと呼び，13 トリック連続して行うことで 1 ゲームが終了する．また、各トリックで最初に出されるカードのことをリーディングカードと呼び，そのカードを出すプレイヤを親と呼ぶ．親以外のプレイヤを子と呼ぶ．子は，リーディングカードと同じスートを場に出す．同じスートのカードがない場合のみ，任意のカード出すことができる．トリック終了時に，リーディングカードと同じスートで最も強いカードを出したプレイヤが，場のカードを全て引き取り，次のトリックの親となる．1 ゲーム終了後，各プレイヤの引き取ったカードのうち，ハートは 1 枚 1 点，スペードの Q は 1 枚 13 点として各プレイヤの罰点を計算する．罰点を少なくすることが各プレイヤの目的である．本研究では，一般的に用いられる「最初のトリックの前に行う手札交換」と「シュート・ザ・ムーンを含む得点のバリアント」の 2 つのルールを無視した．

2.2 対戦に用いるエージェント

本研究では，学習エージェントの学習及び評価実験における対戦相手として， ルールベースエージェント, モンテカルロエージェントの 2 種類のエージェントを用いた．ルールベースエージェントは gnome-hearts[9] を元にしたルールを搭載しているエージェントである．局面情報を考慮して，設定された if - then ルールに基づき出し札を決定する．モンテカルロエージェントは UCT モンテカルロ法 [10] により FALCON に設定した行動種類から行動を選択するエージェントである．行動の探索は，現在トリックのみをシミュレート数 200 回とした．事前に行った実験により，これら 2 種類のエージェントの中ではモンテカルロエージェントが最も強く，ルールベースエージェントが最も弱いということが分かっている [5][6].

3 FALCON による強化学習

3.1 FALCON の構成

図 1 に，FALCON の構成図を示す．sensory field (SF), motor field (MF), feedback field (FF) はそれぞれ cognitive field(CF) と関連付けられている．エージェントが M 個の知覚センサを有するとき，SF には知覚ベクトル $S = (s_1, ..., s_M)$ が与えられる．MF には行動ベクトル $A = (a_1, ..., a_K)$ が与えられる．K は選択可能な行動の数を表し，$a_i (i = 1, ..., K)$ の値は i 番目の行動が選択されたときに 1，それ以外の場合は 0 となる．FF には報酬ベクトル $R = (r, 1-r)$ が与えられる．r はエージェントが環境から受け取る報酬であり，$r \in [0, 1]$ である．CF は L 個のニューロン $n_j (j = 1, ..., L)$ を有し，SF, MF 及び FF のニューロンとそれぞれ重みベクトル $W^s{}_j = (w^s{}_{1j}, ..., w^s{}_{Mj})$, $W^m{}_j = (w^m{}_{1j}, ..., w^m{}_{Kj})$ 及び $W^f{}_j = (w^f{}_{1j}, w^f{}_{2j})$ によって関連付けられている．重みベクトルの要素 $w^y{}_{xj} \in [0, 1]$ は，ニューロン n_j と SF, MF 及び FF のニューロンとの関連の強さを表す．FALCON では，行動選択フェーズと学習フェーズと交互に行うことで学習が進行する．行動選択フェーズでは，知覚情報から最適と思われる行動を選択する．学習フェーズでは，環境からのフィードバックを用いて，重みベクトル更新し，知覚，行動，報酬の関連を学習する．

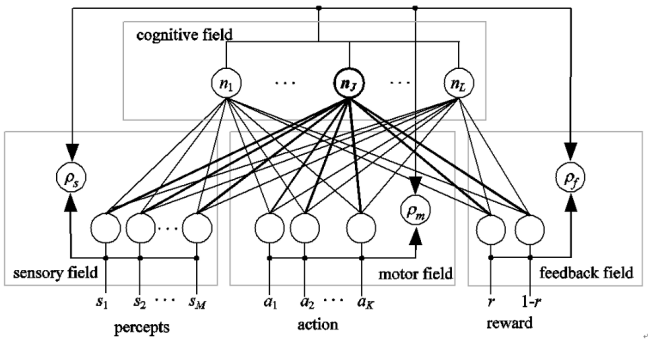

図 1: FALCON の構成.

3.2 FALCON を用いたハーツのための学習

FALCON を用いて学習を行うためには，行動種類と知覚情報を設定する必要がある．本研究ではハーツによる実験を行うため，行動種類は出し札を決定するための手続き，知覚情報は局面状態とした．知覚情報の種類及び，学習における重みの更新方法は我々の先行研究 [6] と同様なものを用いた．学習において，FALCON を用いた学習エージェントは，選択可能な行動種類の中から重みの値が最大である行動を選択して実行する．そして環境からフィードバックを受け取り，行動ベクトルの値を更新することで学習が進行する．また，我々が先行研究で提案してトリック別，かつ親子別にそれぞれ独立した 52 個の FALCON を用いて学習を行う手法 [5] を適用した．

4　サポートベクター回帰

2 値分類問題を解くために Vapnik らによって提案された教師あり学習によるパターン認識モデルの一つとして，サポートベクターマシン (Support Vector Machine: SVM) がある [7]．また，SVM を回帰に適用し，回帰関数を学習する手法をサポートベクター回帰 (Support Vector Regression : SVR) という．本研究では， SVR を適用するにあたりオープンソースの機械学習ライブラリの一つである『LIBSVM』(Version 3.22)[8] を利用した．SVM 及び SVR に関する複数の手法が提供されているが，実験では ν-SVR という手法を利用した．

5　提案手法

5.1　行動種類の変更

我々の先行研究 [5][6] では，罰点札である ♠Q と ♡ のカードや，手札内でのカードの強弱関係を考慮した行動種類を設定していた．こうすることで，ハーツにおける基本的な行動選択規則を獲得することができ，ルールベースエージェントを打ち負かすことに成功した．しかしながら，モンテカルロシミュレーションにより行動を決定するモンテカルロエージェントには勝つことができなかった．その原因は，人間のような戦略的な行動規則, 特に駆け引きを伴うような行動規則を獲得することができなかったためであると考えられる．そこで，実世界のハーツにおける戦略について調査し，学習エージェントの行動種類に取り入れた．本研究では，ハーツの戦略を基にして次の 5 つのテクニックを採用した．(1)Void:あるスートのカードが手札に 1 枚も無い状態になるようカードを出すプレイ．(2)Smoke Out:同じスートを何度もリードし，特定のカードを出すように誘導するプレイ．(3)Cash:強いカードを出し意図的にトリックを取るプレイ．(4)Exit:弱いカードを出しトリックを取らないようにするプレイ．(5)Spear Play:♠Q でリードするプレイ．

本研究で用いた行動種類の一部を表 1 に示す．表 1 において，親番の行動は par1〜par18 に対応しており，子番のリーディングカードと同じスートのカードが手札にある場合は ch1〜ch5 に，ない場合は ch'1〜ch'6 にそれぞれ対応している．

表 1: 設定した行動の一部.

ID	Actions
par1	Play a card of suit ♠.
par2	Play the strongest ♠ in weaker cards than ♠Q.
par3	When the number of agent's cards ♡ is 2 or less, play the strongest ♡.
par4	When the number of agent's cards ♣ is less than average, play the strongest ♣ in weaker cards than strongest ♣ which opponents possess.
par5	It is the same as par4 about ♢.
par6	It is the same as par4 about ♠.
par7	Play the strongest ♡ in weaker cards than strongest ♡ which opponents possess.
…	…
par15	Play the strongest ♡.
…	…
ch1	Play a stronger card of ♠K and ♠A if there is even one of ♠K and ♠A in hands and ♠Q has been played.
ch2	Play the strongest card in weaker cards than the strongest card that is the same as suit of leading card.
ch3	Play the strongest card.
ch4	When the agent has cards of 7 to J, play a card with priority from the smaller one.
ch5	Play the weakest card.
…	…
ch'4	When the number of agent's cards ♢ is 2 or less, play the strongest ♢.
ch'5	It is the same as ch'4 about ♣.
ch'6	♡, Play the strongest card with priority in the order of ♠, ♢, ♣.

5.2　SVR を適用した FALCON

5.2.1　SVR を適用した FALCON の構成

先行研究における FALCON の学習において，ゲーム開始から弱いカードを優先して出し，序盤で弱いカードを出しきってしまうために終盤でゲームが不利になる傾向が多く見られた．これはゲームの性質上，弱いカードを優先的に出すという行動が罰点を取りにくいために正のフィードバックを受け取りやすく，そのような行動が選ばれるように学習が進むためである．そこで本研究では，カードの強弱のみで行動を決めるのではなく，現在の局面情報に対する各出し札の将来的な危険度を SVR を用いて推定し，それを用いて行動を決定する手法を提案する．ここで将来的な危険度とは，現在以降のトリックにおいて ♠Q を獲得するか否かを表す尺度 (以下，DegSQ と表す) である．SVR を適用した FALCON の構成を図 2 に示す．SVR を適用した FALCON では，通常の FALCON と同様に，はじめに選択可能な行動種類の中から重みの値が最大である行動を選択する．次に選択された行動を SVR によって逐次的に評価し，その行動に対する DegSQ を推定する．具体的には，決定された出し札の情報と FALCON に設定した知覚情報を特徴とし，LIBSVM を利用し SVR により DegSQ を推定する．もし，推定された DegSQ

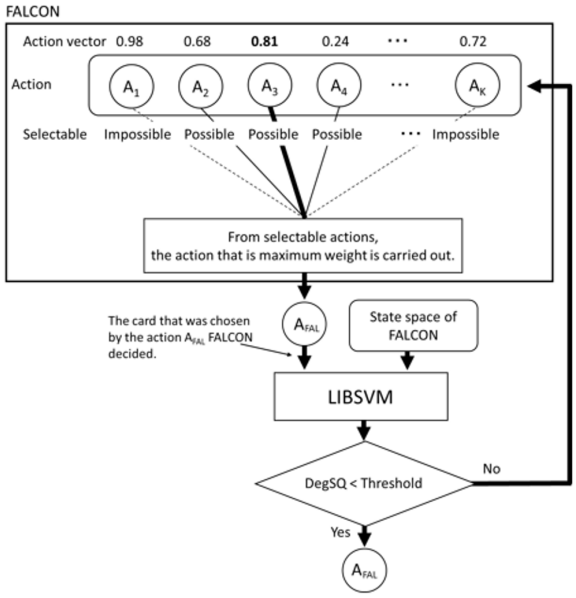

図 2: SVR を適用した FALCON の構成.

があらかじめ設定した閾値よりも小さければその行動を実行し，そうでなければ，DegSQ が閾値よりも小さくなるまで，次に重みの大きな行動を選択し DegSQ を推定するまでを繰り返す．また，DegSQ が閾値よりも小さくなるような行動が存在しない場合は，FALCON が最初に選択した行動を実行する．

5.2.2 SVR 学習モデルの作成

LIBSVM を用いて学習を行うためには，ラベルと特徴から成る訓練用データセットを用意する必要がある．モンテカルロエージェント 4 体を 100,000 ゲーム対戦させ対戦ログを記録し，それを用いて訓練用データセットを作成した．ラベルは，あるトリックにおいて，そのトリック以降に ♠Q を獲得する場合は 1, しない場合は-1 とした．つまり，DegSQ が 1 に近いほど将来的に ♠Q を獲得する確率が高く，-1 に近いほど獲得しない確率が高いといえる．また特徴は，出し札のスートとランクを表す 5 次元と，ゲーム局面を表す 22 次元の計 27 次元のベクトルを用いた．ゲーム局面を表す特徴は，FALCON に与えられる知覚情報と等しい．データセットは手番及びトリック別のグループに分け生成した．また，各グループに対して，ラベルごとに 10,000 件のデータセットを収集した．ただし，ゲームの性質上起こりにくい局面が存在するため，データセット数が 10,000 件に満たない場合はデータセット数が少ないラベルに合わせてアンダーサンプリングして用いた．LIBSVM には，データセットをスケーリングするためのプログラムが提供されている．このプログラムを用いて生成したデータセットをスケーリングし，スケーリング後のデータセットを訓練用データセットとして用いた．また，学習には ν-SVR を使用し，ν-SVR のパラメータ値は，予備実験により決定した．

6 実験

5 章で紹介した提案手法の有効性を確かめるために，2.1 節で紹介したルールに基づくハーツの対戦実験を行った．エージェントの評価方法として，ハーツにおける強さ，つまり平均獲得罰点比率の低さを比較する．平均獲得罰点比率とは，4 体のエージェントの合計罰点 (1 ゲームあたり 26 点) に対するエージェントの獲得罰点の比率である．我々の先行研究 [5] において最良の結果を示した状態のプログラムに対し，本研究における提案手法をすべて適用した場合の学習実験の方法を以下に示す．まず，学習エージェントをルールベースエージェント 3 体と N ゲーム対戦させ学習させる．学習後，学習エージェントをルールベースエージェント 3 体及びモンテカルロエージェント 3 体と 5,000 ゲーム対戦させ平均獲得罰点比率を測定する．ただし，学習後の行動選択において SVR は使用しない．これを 5 回繰り返して行い，平均獲得罰点比率を算出する．

6.1 学習ゲーム数による比較

N の値を 0, 1, 100, 1,000, 10,000, 以降 10,000 刻みで 100,000 まで変化させ，提案手法を適用する場合としない場合それぞれについて実験を行った．図 3 に学習後にルールベースエージェント 3 体と対戦した場合の，図 4 に学習後にモンテカルロエージェント 3 体と対戦した場合の学習エージェントの平均獲得罰点比率をそれぞれ示す．

図 3 及び図 4 より，平均獲得罰点比率は提案手法を適用した場合の方が低くなっていることがわかる．ルールベースエージェントと対戦する場合は，先行研究においても学習エージェントの平均獲得罰点比率は 0.25 を下回りルールベースエージェントに勝つことが可能であった．提案手法を適用することで，更に平均獲得罰点比率を減少させることができた．また，モンテカルロエージェントと対戦する場合は，先行研究においては学習エージェントの平均獲得罰点比率が 0.25 を下回ることはできなかった．提案手法を適用することで，平均獲得罰点比率を減少させることができ，0.25 を下回り勝利する場合も見られた．

6.2 手法による比較

本研究では大きく分けて 2 つの手法を提案したが，この 2 つの提案手法がそれぞれどのくらい学習エージェ

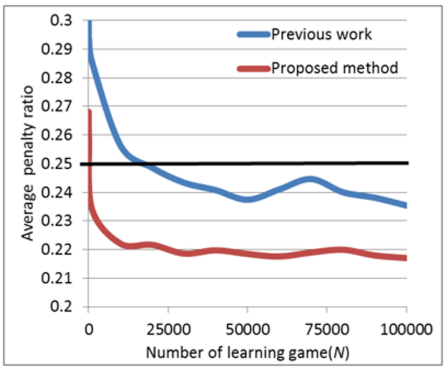

図 3: 学習後にルールベースエージェント 3 体と対戦させた場合における学習ゲーム数 N と学習エージェントの平均獲得罰点比率の関係.

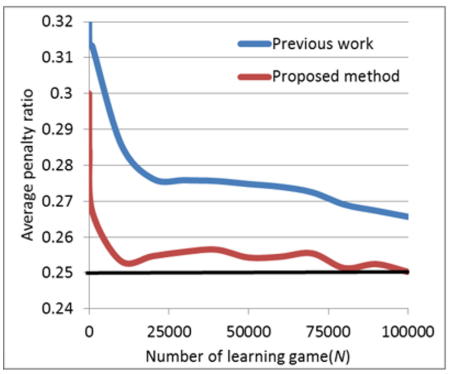

図 4: 学習後にモンテカルロエージェント 3 体と対戦させた場合における学習ゲーム数 N と学習エージェントの平均獲得罰点比率の関係.

ントの性能向上に貢献しているかを確認するための実験も行った．我々の先行研究 [5] のプログラムに対し行動種類の変更のみを行った場合と，SVR の適用のみを行った場合それぞれにおいて，上記と同様な実験を行った．そして，学習ゲーム数が 100,000 ($N = 100,000$) であるときの，学習後の学習エージェントの平均獲得罰点比率を表 2 にまとめる．また，表 2 中の括弧内は，5 試行中に学習エージェントの平均獲得罰点比率が 0.25 を下回り，対戦相手のエージェントのものよりも低かった回数である．

表 2 より，行動種類の変更のみを適用した場合は適用前後で平均獲得罰点比率が減少していることが分かる．一方で，SVR の適用のみを行った場合は適用前後で平均獲得罰点比率に差はあまり見られない．よって，行動種類を変更したことで，より応用的な行動選択規則を獲得することができるようになり，平均獲得罰点比率を減少させることができたと考えられる．しかし，それだけではモンテカルロエージェントに勝つことはできず，更に SVR を適用したことでモンテカルロエージェントに匹敵する学習結果が得られ，5 試行中 2 試行で，0.25 を下回った．これは，SVR を適用したことで学習中に選択される行動種類が変化し，より適した行動選択規則を獲得することができたためであると考えられる．よって，提案手法を適用することは，学習エージェントの平均獲得罰点比率を減少させる上で有効であると言える．

表 2: 学習ゲーム数が 100,000 の場合における学習後の学習エージェントの平均獲得罰点比率.

	Against Rule-based agents	Against Monte-Carlo agents
Previous work	0.2354(5)	0.2657(0)
Only changing action set	0.2181(5)	0.2549(1)
Only applying SVR	0.2429(5)	0.2685(0)
Applying both	0.2170(5)	0.2503(2)

6.3 順位による比較

本節では，学習エージェントの平均獲得罰点比率ではなく，4 プレイヤ間の順位による評価を行う．実験方法としては，学習ゲーム数 N を 100,000 とし，ルールベースエージェント 3 体と対戦し学習させる．学習後，ルールベースエージェント 3 体及びモンテカルロエージェント 3 体と 1,000 ゲーム対戦させ平均順位を計測する．ここでは，学習エージェントを含めた 4 プレイヤの内，いずれかのプレイヤの合計獲得罰点が 100 点に達するまでを 1 ゲームとする．これを 5 試行行い，学習エージェントの平均順位を算出する．我々の先行研究 [5] 及び提案手法に対する順位による評価をそれぞれ表 3，表 4 に示す．

表 3 及び表 4 より，ルールベースエージェント，モンテカルロエージェントの両エージェントに対して，提案手法を適用した学習エージェントの平均順位が高いことが分かる．提案手法において，ルールベースエージェントに対しては，1 位率が 4 割近い値となっている．また，モンテカルロエージェントに対しては，平均順位の期待値が 0.25 に近い値となっており，モンテカルロエージェントとほぼ同等の性能にまで向上していることが分かる．よって，順位による評価からも，提案手法の有効性を示していると言える．

7 むすび

本研究では，ファジィ ART を拡張した機械学習手法である FALCON を用いたハーツ用の自律エージェントの性能を向上させるために，行動種類の改良手法と，

表 3: 先行研究に対する，学習ゲーム数が 100,000 の場合における学習後の学習エージェントの順位.

Rank	Against Rule-based agents	Against Monte-Carlo agents
1	0.2884	0.1808
2	0.2572	0.2270
3	0.2448	0.2554
4	0.2096	0.3368
Expected value	2.3756	2.7482

表 4: 提案手法に対する，学習ゲーム数が 100,000 の場合における学習後の学習エージェントの順位.

Rank	Against Rule-based agents	Against Monte-Carlo agents
1	0.3906	0.2446
2	0.2706	0.2534
3	0.2126	0.2444
4	0.1262	0.2576
Expected value	2.0744	2.5150

SVR を適用した行動選択手法を提案した．また，提案手法の有用性を確認するために対戦実験を行った．

提案手法を適用することで，我々の先行研究よりも学習エージェントの平均獲得罰点比率が減少することを確認した．特に，先行研究における学習エージェントはモンテカルロエージェントに勝つことができなかったが，本研究の提案手法を適用することにより，学習エージェントがモンテカルロエージェントを打ち負かすことがあることも確認することができた．また，ルールベースエージェントに対する平均獲得罰点比率を更に減少させることもできた．よって，提案手法は学習エージェントの性能を向上させる上で有用だといえる．

今後の課題としては，SVR 学習モデルの分類性能を向上させることが挙げられる．訓練用データセットに用いる特徴や，LIBSVM のパラメータを調整することで，予測性能を向上させることができると考えられる．また本研究では，最も大きな罰点札である ♠Q に関する予測のみを行ったが，罰点札である ♡ の各カードなど他の要素に関する予測も行うことで，より戦略的な行動選択規則を獲得することができると期待できる．

謝辞

本研究は広島市立大学特定研究（一般研究費）による支援を受けた．

参考文献

[1] 星野准一: 特集「ゲーム AI」の企画にあたって, 人工知能誌, vol. 23, no. 1 (2008)

[2] Ah-Hwee Tan: FALCON: A Fusion Architecture for Learning, COgnition, and Navigation, *International Joint Conference on Neural Networks*, vol. 4, pp. 3297–3302 (2004)

[3] Dan Xiao, Ah-Hwee Tan: Scaling up Multi-Agent Reinforcement Learning in Complex Domains, *International Conference on Web Intelligence and Intelligent Agent Technology*, vol. 2, pp. 326–329 (2008)

[4] Di Wang, Budhitama Subagdja, Ah-Hwee Tan, and Gee-Wah Ng: Creating human-like autonomous players in real-time first person shooter computer games, *Twenty-First Innovative Applications of Artificial Intelligence Conference*, pp. 173–178 (2009)

[5] Kenta Nimoto, Kenichi Takahashi, and Michimasa Inaba: Improvement of Agent Learning for a Card Game based on Multi-channel ART Networks, *JOURNAL OF COMPUTERS*, vol. 11, no. 4, pp. 341–352 (2016)

[6] Kenta Nimoto, Kenichi Takahashi, and Michimasa Inaba: Construction of a Player Agent for a Card Game Using an Ensemble Method, *Procedia Computer Science*, vol. 96, pp. 772–781 (2016)

[7] Vladimir N. Vapnik and Aleksandr Ya. Lerner: Pattern recognition using generalized portrait method, *Automation and Remote Control*, Vol. 24, No. 6, pp. 774–780 (1963)

[8] Chih-Chung Chang and Chih-Jen Lin: LIBSVM – A Library for Support Vector Machines, *Procedia Computer Science*, `https://www.csie.ntu.edu.tw/~cjlin/libsvm/`, 参照 Jan. 27, 2017.

[9] Lone Wolves: Lone Wolves-Web, game, and open source development, `https://launchpad.net/ubuntu/+source/gnome-hearts`, 参照 Jun. 13, 2016.

[10] Levente Kocsis, Csaba Szepesvari: Bandit Based Montecarlo Planning, *European Conference on Machine Learning*, 参照 Jan. 27, 2017.