行動価値に着目した学習分類子システムの改善：マルチエージェント強化学習への接近

井上寛康[†] 高玉圭樹[†,††] 下原勝憲[†,†††]

これまでで最も改善された学習分類子システムである XCS は，決定的状態遷移からなる環境でのみ正しく動作することが知られている．本論文では，決定的状態遷移環境よりも複雑なマルチエージェント環境でも利用できる学習分類子システムを目指し，適切な経験の一般化が可能な XCS-QT を提案する．そしてその優位性をシミュレーション実験により示す．具体的には木の問題および追跡問題を用いて実験し，マルチエージェント環境は XCS にとって正しく動作できないいくつかの要因が含まれていること，および XCS-QT がそれら要因を克服することを示す．

Improvement of Learning Classifier System by Action-value Function toward Multi-agent Reinforcement Learning

Hiroyasu Inoue,[†] Keiki Takadama[†,††] and Katsunori Shimohara[†,†††]

XCS is the newest Learning Classifier System (LCS), and at present it can only be used for deterministic transition environments. This paper proposes XCS-QT as a modified LCS that can appropriately generalize its experience and can be used for multi-agent environments that are more complex than deterministic transition environments. We then show the system's advantage via simulation experiments using quasi-tree problems and hunter problems. Through the experiments, we demonstrate that there are several reasons why XCS cannot work very well in multi-agent environments, and that XCS-QT can overcome those problems.

1. はじめに

環境との相互作用および報酬を手がかりとし，エージェントが適応するためのシステムとして，学習分類子システム（Learning Classifier System）[1] が提案されている．同様のシステムとして強化学習手法[2] があるが，LCS は強化学習手法にはない強力な経験一般化の機能を保持している．この経験一般化は広大な状態空間が存在する場合に不可欠であるが，LCS では進化手法を用いたルール発見という形で簡便に実現されている．

Holland が提案した LCS は，その後 ZCS [3] としてより簡易な形で表現され，さらに一般化の機能が適切に発揮されるように改善された XCS [4] へと発達している．XCS は分類子の適応度を導入し，優れた一般化を実現している．しかしながら，この XCS は環境が決定的状態遷移からなるときにのみ適切な一般化が行われることが分かっており，それより複雑な環境では適切な一般化ができない．

ところで，マルチエージェントシステムにおけるエージェントの設計は複雑であるため，学習，とりわけ，陽な環境モデルを必要とせず，遅れ報酬から学習できる強化学習への期待は大きい[5]．さらに，LCS は強力な経験一般化の機能を保持しているため，これをマルチエージェント強化学習で利用できることは意義がある．しかしながら，マルチエージェント強化学習において環境が決定的状態遷移からなることは現実的にはありえない．

マルチエージェントシステムに LCS を適用した研究は多くある[6)~8)]．これらの研究は LCS を適用した結果について議論している．しかしながら，LCS が適切に動作しているか議論した研究はほとんどない．

上記をふまえ本論文では，決定的状態遷移環境よりも複雑なマルチエージェント環境においても，XCS に

† ATR ネットワーク情報学研究所
ATR Network Informatics Laboratory
†† 東京工業大学大学院総合理工学研究科
Interdisciplinary Graduate School of Science and Engineering, Tokyo Institute of Technology
††† 京都大学情報学研究科
Graduate School of Informatics, Kyoto University

おける分類子の適応度が正しく求まるように改善を行う．具体的には XCS に Q-Table を付加した XCS-QT を提案し，XCS よりも優れた性能を持つことを実験により示す．実験では木状の環境を用い，決定的状態遷移環境からマルチエージェント環境に至る困難さを段階に分けた，決定的状態遷移環境，確率的状態遷移環境，部分知覚環境，マルチエージェント環境における性能の違いを検証する．

以下，2 章では LCS，3 章ではマルチエージェント強化学習における困難さについて述べ，4 章で新たな XCS を提案する．5 章では木の問題での実験と議論，および 6 章では追跡問題での実験と議論を行う．最後に 7 章で結論を述べる．

2. 学習分類子システム

2.1 学習分類子システムとは

学習分類子システム[1]（Learning Classifier System）は環境との相互作用を通じて分類子と呼ばれる条件–行動ルールを学習する．この学習は，Q 学習や Sarsa などの強化学習手法[2] と同様に行われる．具体的には，環境における現在の状態に応じて行動を実行し，その結果得られた報酬に基づいて学習が行われる．さらに LCS においては遺伝的アルゴリズム[9] を用いて分類子の生成と削除をすることにより，システムが保持する分類子の集合をより環境に適したものに変更する．

ところで，強化学習手法には一般化の機能が付加されることがある．具体的に一般化とは，すべての状態での行動を実際に経験し学習することは困難であるため，ある経験を一般化し他の状態での同じ行動も経験したとすることである．これによって，環境の状態が増えるに従って学習が困難になる次元の呪い[10] に対応することができる．

LCS はこの**一般化**の機能が簡便に実現されている．具体的には，ルールの条件部にワイルドカードを含むことにより，複数の状況に適合する分類子により実現される．

ある経験をどの範囲の状態での経験とするかという一般化の範囲は，強化学習では一般的に固定である．一方で，LCS は上述の遺伝的アルゴリズムの効果により，その範囲が動的に変化していく．上記をふまえると，強化学習手法と比較して LCS は以下のメリットがある．

(1) 動的な一般化の機能は，強化学習手法では実現困難であるが，後述するように LCS では容易である．

(2) 動的な一般化によって最終的に得られたルールは可読性があり，人がシステムを分析する際に役に立つ．

これらのメリットをマルチエージェント強化学習においても享受できるようにすることが，本論文の狙いである．

LCS はこれまでに改善が行われているが，本論文では，**適応度を持つ LCS（XCS）**[4] を用いる．この理由は，これ以前の LCS は一般化の能力が適切でないことが指摘されており[11]，この XCS が現在の主流のためである．

2.2 適応度を持つ学習分類子システム（XCS）

ここでは適応度を持つ LCS を**図 1** に基づいて説明する．XCS は環境からの入力である状態（State）に対して適切な行動（Action）を実行することで環境から得られる（Reward）の合計を最大化するように学習する．図 1 の検出器（Detector）は環境の状態を XCS の内部表現に，効果器（Effector）はその逆に変換する．

以下では（1）分類子の形式，（2）実行部，（3）強化部，（4）発見部の順に XCS を説明する．

2.2.1 分類子の形式

図 1 の分類子群（Classifier Population）における分類子は条件部と行動部の対からなる．条件部は離散値あるいは任意の値を示すワイルドカード（以降#）によって表現される．たとえば 1#0 などは 3 つの条件からなる条件部の例である．行動部は任意の表現でよい．また，同じ条件部と行動部を持つ分類子を 1 つの分類子としてまとめるために，その重複数（Numerosity）を保持する．分類子にはほかに，予測値（Prediction），誤差値（Error），適応度（Fitness）を持つ．以下では，

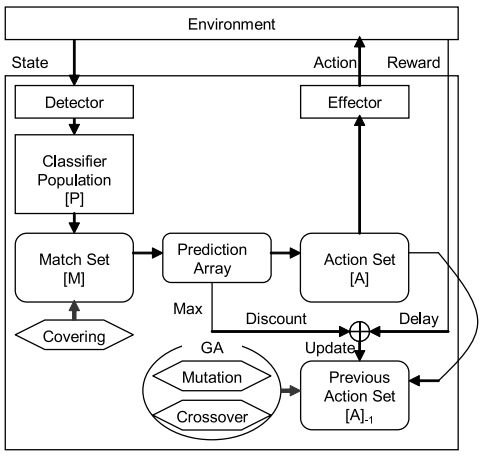

図 1 XCS
Fig. 1 XCS.

ある i 番目の分類子を cl_i とし，その重複数，予測値，誤差値，適応度をそれぞれ num_i，p_i，ϵ_i，F_i とする．

分類子群には最大の分類子数（N）の制限があり，初期状態では条件部と行動部をランダムに生成した N 個の分類子を持つ．その際の p_i，ϵ_i，F_i の初期値を PI，EI，FI とする．

2.2.2 実 行 部

実行部は分類子群を参照し，状態に対して適切な行動を選択し，出力する過程を担う．環境から状態が入力されると，図 1 における [P] で表された分類子群に対して条件部と状態の照合が行われる．その照合では，#を除いてまったく一致する分類子かどうかが調べられる．照合で一致した分類子は図 1 における [M] で表された照合群（Match Set）を形成する．なお，照合時に分類子群に一致する分類子が存在しない場合，入力された状態に一致する分類子を新たに生成する被覆（Covering）メカニズムが実行される．この被覆メカニズムでは，入力された状態に一致するように条件部が作成されるが，そのとき各条件に#をある確率（$P_\#$）で含むように作成される．

照合群内の分類子行動部には複数の行動が含まれて競合することがある．これを解消するために，競合する行動による予測利益を連ねた予測配列（Prediction Array）が作成される．予測配列における各行動 a_i の予測利益 P_i は次のように計算される．

$$P(a_i) = \frac{\Sigma_{cl_k \in [M]|a_i} p_k \times F_k}{\Sigma_{cl_k \in [M]|a_i} F_k} \tag{1}$$

ただし，$[M]|a_i$ は照合群内の行動部に行動 a_i を持つ分類子の集合を表す．この予測配列に基づいて行動選択が行われるが，ここでは強化学習手法で一般的に用いられている ϵ グリーディ戦略をとり，その ϵ を P_{explore} とする．選ばれた行動を行動部に持つ分類子によって図 1 における [A] で表された行動群（Action Set）が形成される．そして，選択された行動が実行されることで実行部の一連の処理が終了する．この一連の処理をステップと呼ぶ．

利益の予測は，式 (1) にあるように単に予測値のみではなく，適応度との積であるのが重要である．もしこのような適応度がなければ，#を含んで正の報酬の直前で選ばれる分類子があった場合，その分類子の予測値は大きいため，他の状態でも選ばれることで，適切な行動を妨げる．このような事態は適応度を導入した XCS では発生しない．

2.2.3 強 化 部

強化部は各ステップにおける実行部の処理が完了した後に，行動群に含まれる分類子の予測値，誤差値，適応度を，環境から得られる報酬に基づいて更新する．この過程は図 1 における $[\mathrm{A}]_{-1}$ で表された 1 ステップ前の行動群（Previous Action Set）に対して行われる．その更新は，各分類子 cl_j に対して次の更新規則が適用することで実現される．なお左矢印は代入を意味する．

$$P \leftarrow r + \gamma \max_a P(a) \tag{2}$$

$$p_j \leftarrow p_j + \beta(P - p_j) \tag{3}$$

$$\epsilon_j \leftarrow \epsilon_j + \beta(|P - p_j| - \epsilon_j) \tag{4}$$

ただし，P は予測値 p_j を更新する際の目標値であり，1 ステップ前の報酬 r と予測配列中の最大値 $\max_a P(a)$ を用いて計算される．β と γ は強化学習手法における学習率と割引率と同じであり，それぞれ適応の早さと，将来の報酬をどれぐらい考慮するかを表す．

そして，次式に示すように更新後の誤差値 ϵ_j に基づいて分類子の絶対的な適応度 κ_j と $[\mathrm{A}]_{-1}$ における相対的な適応度 κ'_j が順に計算され，最期に適応度 F_j が計算される．

$$\kappa_j = \begin{cases} 1 & \text{if } \epsilon_j \le \epsilon_0 \\ \alpha(\epsilon_j/\epsilon_0)^{-\nu} & \text{otherwise} \end{cases} \tag{5}$$

$$\kappa'_j = \frac{(\kappa_j \times num_j)}{\Sigma_{cl_k} \in [A]_{-1}(\kappa_k \times num_k)} \tag{6}$$

$$F_j \leftarrow F_j + \beta(\kappa'_j - F_j) \tag{7}$$

ここまでが強化部における一連の処理である．

2.2.4 発 見 部

発見部では，遺伝的アルゴリズムを用いた分類子の生成と削除を通して，分類子群を進化させる．遺伝的アルゴリズムは 1 ステップ前の行動群 $[\mathrm{A}]_{-1}$ の各分類子において，前回遺伝的アルゴリズムが実行されてから経過したステップ数の平均が特定の値（θ_{GA}）の値を上回った場合に実行される．

遺伝的アルゴリズムが実行されると，$[\mathrm{A}]_{-1}$ の各分類子の適応度の大きさを相対的な選択確率として 2 つの分類子が親個体として選択される．そして，親個体に対して交叉（Crossover）および突然変異（Mutation）がそれぞれある確率（χ，μ）で実行される．生成された 2 つの子個体は分類子群に追加される．そのとき，最大数 N を超えた場合は適応度に応じて削除される☆．この発見部により分類子が生成・削除されるが，このことで条件部に存在する一般化の適切な範囲を探索していることに注意されたい．

☆ アルゴリズムは文献 12) を参照されたい．

3. マルチエージェント強化学習の困難さ

本論文では，マルチエージェント強化学習に対して LCS を利用することを目指しているが，2.2 節で述べた XCS は，ある状態でのある行動の遷移先は決定的であるという前提を要求している．これは強化学習手法が一般的に前提としている有限マルコフ決定過程が，確率的な状態遷移であることから考えると部分クラスとなる．また，マルチエージェント強化学習は有限マルコフ決定過程ではなく，非マルコフ決定過程である．本章では，このような環境に XCS を利用することでどのような問題が発生するかを，決定的状態遷移環境，確率的状態遷移環境，部分知覚環境，そしてマルチエージェント環境の順に説明する．

3.1 決定的状態遷移環境

図 2 の左上は決定的状態遷移環境における例を示している．図のように状態 s で行動 a をとると，必ず状態 s' に向かうため，この行動を実現した行動群には式 (2) で示したように，次の行動時の予測配列値に割引率をかけたものと報酬の和が目標の予測値として与えられる．

行動群に含まれる分類子のうち，**#を含まない**分類子に与えられる目標の予測値は緩やかに変化する．なぜなら，目標の予測値は引き続く行動群の予測値にのみ依存するためである．このことは式 (4) で得られる誤差値は小さいことを意味し，発見部で淘汰されない．

行動群に含まれる分類子のうち，**#を含む**分類子は状態 s だけでなく複数の異なる他の状態でも利用される．その際に，目標の予測値がそれらの状態で大きく異なれば誤差値が大きく，その分類子は式 (1) にあるように，予測配列における発言力を失い，かつ発見部により淘汰されうる．すなわち，適切な#を持つ分類子が生き残る．

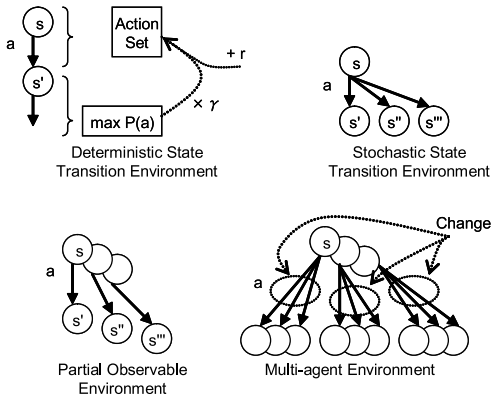

図 2 各状態遷移の様子
Fig. 2 Schemes of state transition.

決定的状態遷移環境においては，XCS は適切な一般化および学習を上述のように行っている．

3.2 確率的状態遷移環境

図 2 の右上は確率的状態遷移環境の例を示している．図のように状態 s で行動 a をとると，状態 s'・s"・s"' などに特定の確率で向かうため，この行動を実現した行動群に対する目標の予測値は，それぞれの次状態から与えられることになる．

この場合，行動群に含まれる分類子のうち，#を含まない分類子に与えられる目標の予測値は，複数の状態から与えられる．それらの目標の予測値は一般的に異なるため，誤差値は大きくなる．#を含まない分類子でも誤差値が大きくなるので，#を含む分類子においては#の含まれ方にかかわらず誤差値が大きくなる．確率的状態遷移環境における上記のような過程により，XCS では一般化と学習を果たすことができない．

3.3 部分知覚環境

エージェントの知覚の能力によって，実際には異なる環境が区別できない場合があり，これを本論文では部分知覚環境と呼ぶ．図 2 の左下は部分知覚環境における状態遷移の例を示している．この図において，区別できない 3 つの状態を s と知覚していることを示しており，行動 a をとるとそれぞれの区別できない状態から，状態 s'・s"・s"' に遷移することを示している．すなわち，状態 s で行動 a を実現した行動群に対する目標の予測値は，異なる環境 s'・s"・s"' での行動時の予測配列値からそれぞれ与えられる．しかしながら，これらの目標の予測値は，一般的には異なるため，誤差値は大きくなる．前節の議論と同様に，#を含まない分類子でも誤差値が大きくなるため，#を含む分類子も誤差値が大きくなる．部分知覚環境における上記のような過程により，XCS では一般化と学習を果たすことができない．

3.4 マルチエージェント環境

マルチエージェント環境は，前節までの確率的状態遷移環境，部分知覚環境を内包している．すなわち，まずエージェントは一般的には確率的方策を持つことと，他のエージェントも環境の一部となるので，状態遷移は確率的となる．本論文の XCS においても ϵ グリーディ戦略をとっているため，確率的状態遷移となる☆．次に，1 つのエージェントにすべての状態を把

☆ これは XCS が ϵ グリーディ戦略でなければならないことを意味しない．一般的に強化学習手法を用いると確率的方策であるということである．実際にルーレット戦略なども XCS に用いられている[13]．

握させず，複数のエージェントを利用し補完することがマルチエージェントシステムの一般的な前提であるため，マルチエージェント強化学習においては部分知覚が必ず含まれる．これらをふまえて，図 2 の右下は，状態 s が部分知覚によって実際には複数の状態であり，それらの複数の状態から行動 a によって状態遷移する先は確率的で複数ある様子を示している．

また，マルチエージェント環境においてはマルチエージェント強化学習に特有の問題があり，それをここでは同時学習問題と呼ぶ☆．同時学習問題とは，エージェントが持つ確率的方策は学習によって変化するということである．すなわち，確率的状態遷移の分布が決定的でないことを示している．図 2 における Change はこれを意味している．

以上の議論により，マルチエージェント環境において，XCS が一般化と学習を果たすことができないのは明らかである．

4. 新たな XCS の提案

4.1 行動価値を用いたアプローチ

3 章で述べた XCS の一般化と学習の困難さは，すべて適切な誤差値（およびそれから導出される適応度）の算出が妨げられることに起因している．#を持たない分類子でも誤差値が大きくなるのはその顕著な例である．誤差値の目的は「一般化の不適切さ」を測ることであるから，一般化を行っていない（#を持たない）分類子の誤差値が大きくてはならない．

本論文では，一般化の正しさを適切に測ることでこの問題を解決する．式 (4) から分かるように，誤差値は予測値の更新時の差を目標に更新される．確率的状態遷移環境・部分知覚環境・マルチエージェント環境においては，複数の状態から目標の予測値が与えられる．それらの異なる予測値が与えられるたびに，誤差値の更新を行うことは適切とはいえない．その代わり，目標の予測値の**期待値**を用いることが適切である．この期待値は，強化学習手法における**行動価値**と同じである．ある分類子が，正しい一般化を行っているならば複数の状態における行動価値の間に差がないはずであり，このとき誤差値を小さくするべきである．また，ある分類子が#を持たない場合，ある 1 つの行動価値を目標とするため，誤差値を小さくするべきである．

4.2 行動価値テーブルを持つ XCS（XCS-QT）

4.1 節での議論をもとに，XCS に改善を施した XCS with Q-Table（XCS-QT）を提案する．**図 3** は図 1 からの改善部分だけを示したものである．改善にともなって変更されるのは，式 (3) に相当する部分である．まず，直接 $[\mathrm{A}]_{-1}$ に与えられていた予測配列中の最大値と報酬が，次式のように，行動価値テーブル（Q-Table）の更新に用いられる．

$$Q(s,a) \leftarrow Q(s,a) + \rho(P - Q(s,a)) \qquad (8)$$

ただし，$Q(s,a)$ は状態 s，行動 a における行動価値を，ρ は行動価値テーブルの学習率を表す．この状態 s および行動 a は，$[\mathrm{A}]_{-1}$ における状態 s および行動 a である．$[\mathrm{A}]_{-1}$ に含まれる分類子の更新は，式 (3) から次式のように行動価値テーブルを用いたものに変更する．

$$p_j \leftarrow p_j + \beta(Q(s,a) - p_j) \qquad (9)$$

ただし，$Q(s,a)$ は $[\mathrm{A}]_{-1}$ における状態 s および行動 a における $Q(s,a)$ である．このように行動価値テーブルを介して $[\mathrm{A}]_{-1}$ を更新するため，図 3 において Indirect Update としている．

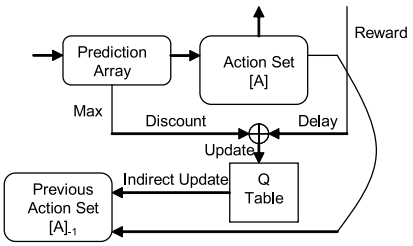

図 3 XCS-QT（一部）
Fig. 3 Part of XCS-QT.

行動価値テーブルを付加した本手法は，行動価値テーブルを持つ一般的な強化学習手法よりも強力である．それは，LCS による動的な一般化の簡便な実現や，学習後のルールの可読性などのためである．

5. 木の問題

3 章で議論した，マルチエージェント環境に含まれる個別の難しさを議論するため，本章では木の問題を取り扱う．

5.1 木の種類

5.1.1 決定的状態遷移環境の木

図 4 は決定的状態遷移をする木を示している．各円の中の数字の列が，知覚器を介して条件部に状態として与えられる．矢印はエージェントの行動 0 と 1 による状態遷移先を示す．左端の Start の状態から始まり，Terminal まで達すると，再び Start に戻る．最後の 2 行動が 1，0 あるいは 0，1 の順であるときのみ最後の行動の際に報酬 1 が与えられ，それ以外では

☆ 他にも多くの問題が存在することが知られている[5] が，ここでは本論文に特に関係ある同時学習問題のみを扱う．

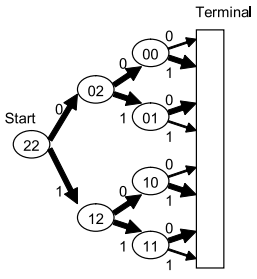

図 4 決定的状態遷移環境の木
Fig. 4 Tree of deterministic state transition environment.

表 1 XCS および XCS-QT のパラメータ
Table 1 Parameters of XCS and XCS-QT.

Parameter	Value	Parameter	Value
N	100	$P_{\#}$	0.33
PI	0.0001	EI	0
FI	0	β	0.01
γ	0.63	α	0.1
χ	0.5	μ	0.01
P_{explore}	0.2	ϵ_0	0.01
ν	0.2	θ_{GA}	50
ρ	0.02		

0 である．図 4 中の太い矢印は報酬 1 が最後に得られるときのパスを示している．

5.1.2 確率的状態遷移環境の木

確率的状態遷移環境の木は，5.1.1 項の決定的状態遷移環境の木における行動が確率的になることを除いて同じである．すなわち確率的状態遷移は，ある確率 P_{rand} で行動がランダムに変更されることで実現される．たとえば図 4 の状態 22 で行動 0 をとった場合，決定的状態遷移ならば状態 02 へ遷移するが，本環境ではエージェントが行動 0 をとったにもかかわらず，状態 12 へ遷移しうる．

5.1.3 部分知覚環境の木

部分知覚環境の木は，5.1.1 項の決定的状態遷移環境の木において，状態を表す数字の列が一部異なることを除いて同じである．その数字の列とは，状態 00，01，10，11 であり，それぞれ状態 00，00，10，10 となる．すなわち，状態 02 および 12 からの遷移は行動 0 と 1 では実際には異なるにもかかわらず，同じ状態として知覚してしまう．これが部分知覚となる．

5.1.4 マルチエージェント環境の木

マルチエージェント環境の木は，5.1.1 項の決定的状態遷移環境の木が 2 体のエージェント用に 2 つ用意される．2 つのエージェントは別々に行動を行うが，最初に Start である状態 22 から始まり，同時に行動を行う．報酬が与えられる条件は 1 体のときと異なり，以下のようになる．

- Start での行動は 2 体のエージェントで互いに異なること．
- Terminal への最後の 2 行動は 0，1 あるいは 1，0 であること．

これらを満たすときのみ 1 が，それ以外では 0 が与えられる．

3 章で述べたように，マルチエージェント環境は確率的状態遷移環境の困難さを内包するので 5.1.2 項のような確率的状態遷移であるべきだと思われるが，3.4 節で述べたように，確率的方策に従ってエージェントが行動しているために，個々の行動による状態遷移が確率的でなくても，状態遷移はすでに確率的である．

5.2 実験設定と結果

XCS および XCS-QT をそれぞれすべての木の環境について実験した．XCS および XCS-QT で用いたパラメータは**表 1** のようになる．また，確率的状態遷移環境においてランダムに状態遷移する確率 P_{rand} は 0.4 である．実験は 100,000 エピソードずつ 10 回行った．

図 5 は 4 つの木における XCS と XCS-QT の実験結果を示している．1,000 エピソードを 1 セットとし，1 セット内のエピソードで得られた報酬を 10 回の実験で平均化している．このようにした理由は，各エピソードでは 0 か 1 かの報酬が得られ，それを各エピソードごとに 10 回の実験で平均したものでは傾向が観察できないほど変動が大きいためである．各グラフの横軸はそれらのセット数を，縦軸はセット内の報酬の平均値を示している．

5.3 議　　論

5.3.1 決定的状態遷移環境

図 5 の左上にあるように，XCS の方が XCS-QT よりも初期の学習が迅速である．これは XCS-QT が Q-Table と分類子とで二重の学習ステップを踏む分，遅れるためである．

1 エピソードあたりの報酬の理論最適値は 0.90 である．これは，最後のステップで適切な行動をとれる確率が P_{explore} によって 0.9 であることに起因する．グラフはその値に収束していることが分かる．

図 4 の太い矢印から分かるとおり，状態 00 と 10 においてはいずれも行動 1 で報酬が得られるため，条件部#0，行動部 1 という分類子は適切な一般化がなされた分類子である．また同様に条件部#1，行動部 0 という分類子も同様に適切な一般化がなされた分類子である．

表 2 の 1 番目に最終的に獲得した分類子を示している．条件部の 1 番目が#である分類子が，予測値，

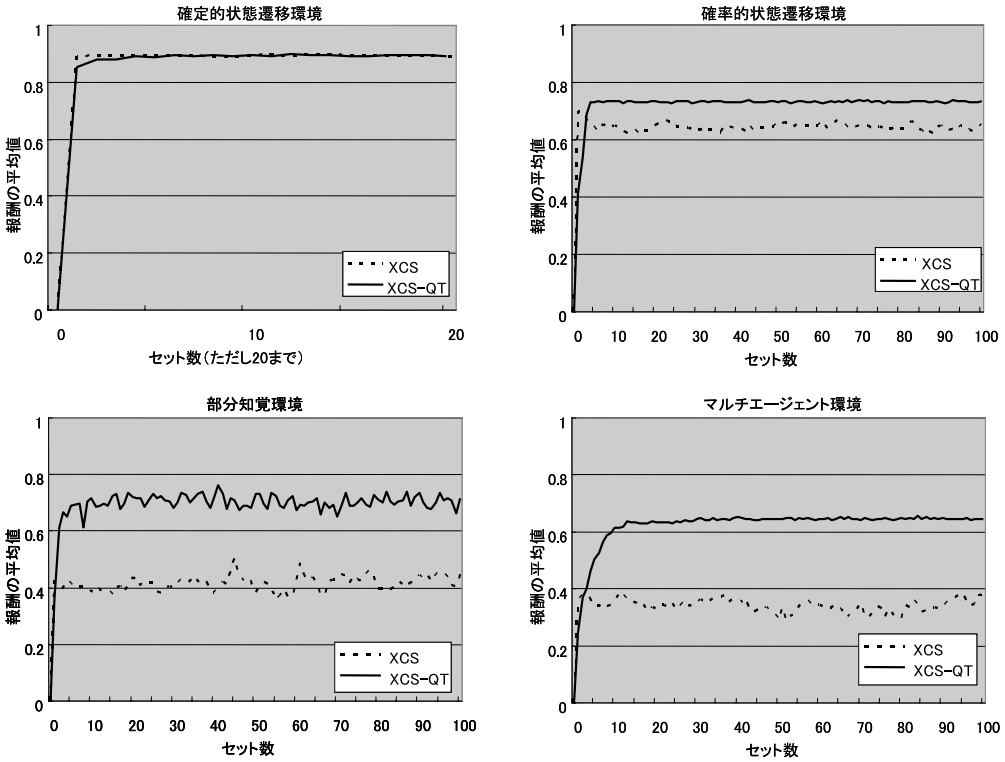

図 5 木の問題の実験結果
Fig. 5 Results of tree problems.

適応度ともに 1.00 で最大の値に収束していることが分かる．これはいずれの手法も適切な一般化ができていることを示す．

5.3.2 確率的状態遷移環境

図 5 の右上にあるように，XCS-QT の方が XCS よりも得られる報酬が大きい．ここで，1 エピソードあたりの報酬の理論最適値は 0.74 である．これは P_{explore} と確率的遷移からなる期待値である．XCS-QT のみがこの値に収束していることが分かる．

表 2 の 2 番目に最終的に獲得した分類子を示している．XCS-QT においては，条件部の 1 番目が#の分類子の適応度は大きく，条件部が##の（役に立たない）分類子の適応度は小さい．一方で XCS においては，条件部の 1 番目が#の分類子と，条件部が##の分類子では適応度の差があまりない．したがって，XCS-QT は適切，XCS は不適切な一般化をしていることになり，これはグラフにおける結果を裏付ける．

5.3.3 部分知覚環境

この環境では図 4 の状態 01 と 11 は存在せず，それぞれ 00 と 10 に替わっている．よって，状態 00 あるいは 10 で行動 0 と 1 をとったときの報酬の期待値は 0.5 ずつである．しかし，もし状態 02 で必ず行動 1 をとるという**偏り**があれば，状態 00 で行動 0 をとることで報酬の期待値は 1 となる．これらのいずれになるとはいえず，理論最適値は求められない．

表 2 の 3 番目に最終的に獲得した分類子を示している．XCS-QT において，（状態 00 にのみ適合する）条件部#0 で行動 0 の分類子は適応度が大きく，行動 1 の分類子は小さい．これは上述したように状態 02 と 12 で行動 1 をとるという偏りを作り出し，それに対応した一般化ができていることを示す．また，条件部が##の不要な分類子は小さい．一方で XCS では，条件部#0 の分類子の適応度は小さく，条件部##の不要な分類子の適応度は大きい．したがって，XCS-QT は適切，XCS は不適切な一般化をしており，図 5 の左下のグラフにおいて，XCS-QT の方が XCS よりも得られる報酬が大きいという結果を裏付ける．

5.3.4 マルチエージェント環境

上述したように部分知覚環境の報酬の理論最適値は求められないため，部分知覚環境を内包するマルチエージェント環境でも求められない．

表 2 の 4 番目に最終的に獲得した分類子を示している．条件部 2#の分類子は状態 22 で用いられるが，XCS-QT において適応度は大きい．そして予測値は

表 2 獲得された分類子例（木の問題）
Table 2 Example of acquired classifiers (tree problem).

決定的状態遷移環境の分類子（ただし一部．p，F は 10 回の実験の平均）

XCS				XCS-QT			
c	a	p	F	c	a	p	F
# 0	1	1.00	1.00	# 0	1	1.00	1.00
# 1	0	1.00	1.00	# 1	0	1.00	1.00

確率的状態遷移環境の分類子（ただし一部．p，F は 10 回の実験の平均）

XCS				XCS-QT			
c	a	p	F	c	a	p	F
# 0	1	0.56	0.37	# 0	1	0.80	0.85
# 1	0	0.32	0.11	# 1	0	0.79	0.80
# #	0	0.39	0.20	# #	0	0.20	0.01
# #	1	0.38	0.20	# #	1	0.15	0.01

部分知覚環境の分類子（ただし一例かつ一部）

XCS				XCS-QT			
c	a	p	F	c	a	p	F
# 0	0	0.60	0.03	# 0	0	0.87	0.91
# 0	1	0.53	0.01	# 0	1	0.24	0.13
# #	0	0.16	0.33	# #	0	0.42	0.04
# #	1	0.14	0.31	# #	1	0.44	0.05

マルチエージェント環境の分類子（ただし一例かつ一部）

XCS の片方のエージェント				XCS-QT の片方のエージェント			
c	a	p	F	c	a	p	F
2 #	0	0.11	0.33	2 #	0	0.04	1.00
2 #	1	0.14	0.46	2 #	1	0.31	1.00
# #	0	0.21	0.29	# #	0	0.11	0.01
# #	1	0.21	0.36	# #	1	0.05	0.01

（ただし c，a，p，F はそれぞれ分類子の条件部，行動部，予測値，適応度を表す）

行動 1 の分類子では大きく，行動 0 では小さい．このことは，（これら分類子を持つエージェントと異なる）もう一方のエージェントの条件部 2#の分類子は逆の傾向を持つことを示しており，このマルチエージェント環境ではより多くの報酬を得るために不可欠である．また，条件部を完全に一般化した不要な分類子の適応度は小さい．一方で XCS では条件部 2#の分類子と条件部##の分類子は適応度においてほとんど差がない．これらのことから，XCS-QT は適切，XCS は不適切な一般化をしており，図 5 の右下のグラフにおいて，XCS-QT の方が XCS よりも得られる報酬が大きいという結果を裏付ける．

6. 追跡問題

5 章では，段階的に問題を複雑にしていくことで XCS-QT の一般化が効果的に作用することを見た．本章では，より複雑で一般的なマルチエージェント強化学習のテストベッドである追跡問題[14] を用いて検証する．

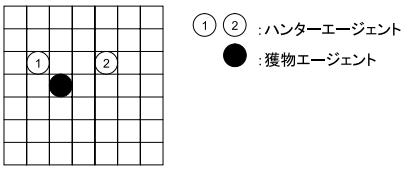

図 6 追跡問題
Fig. 6 Hunter problem.

6.1 追跡問題の規定

図 6 は追跡問題の様子を示している．追跡問題は 7×7 の格子状でトーラスのフィールドを持つ．フィールドには 1 体の獲物エージェント（以下単に獲物）と 2 体のハンターエージェント（以下単にハンタ）が存在する．

エージェントは移動の際，上下左右へ 1 マスか，あるいは動かないを選択する．複数のエージェントは同一のマスに侵入できない．したがって，そのような移動を選択した場合は何も起きない．すべてのハンタが獲物の上下左右のいずれかに隣接したとき，最終状態とする．

獲物は学習を行わず固定された方策で移動する．獲物の視界は獲物を中心に 3×3 であり，この視界内にハンタがいる場合は，それらハンタとのマンハッタン距離の和が大きくなる方向に動き，遠ざかる．視野にハンタがいない場合は動かない．

ハンタは学習を行う．ハンタの視界はハンタを中心に 7×7 である．

報酬は最終状態のときのみハンタに 1 が与えられ，他の状態では 0 が与えられる．獲物が逃避的であることから，2 体のハンタは獲物を挟み撃ちするという協調行動を通信することなく行わなければならない．

6.2 ハンタの検出器

ハンタは他のハンタおよび獲物の位置を知覚する．対象となるエージェントの位置を検出器は以下のように変換する．

- 上下左右の直線上に対象エージェントがいるとき，それぞれその位置を 0，1，2，3 とする．
- それ以外の左上，右上，右下，左下に対象エージェントがいるとき，それぞれその位置を 4，5，6，7 とする．

これらの値を，他のハンタ，獲物の順に並べたものが内部表現となる．図 6 のハンタ 2 の知覚器による内部表現は，2，7 となる．

6.3 追跡問題の流れ

追跡問題は以下を 1 エピソードとして実行される．

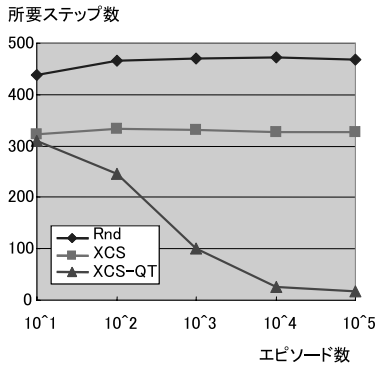

図 7 追跡問題の実験結果
Fig. 7 Results of Hunter problem.

(1) ランダムにハンタと獲物の位置が決められる.
(2) 以下を繰り返す.
  (a) 獲物が移動する.
  (b) ハンタが決められた順番で移動する.
  (c) 最終状態ならば (3) へ.
  (d) ハンタに報酬 0 が与えられ, (a) へ.
(3) ハンタに報酬 1 が与えられる.

6.4 実験設定と結果

XCS および XCS-QT について実験した. パラメータは, N を 1,000, θ_{GA} を 100 とした以外は, 表 1 に従った. 実験は 100,000 エピソードずつ 10 回行った.

図 7 は実験結果を表しており, XCS および XCS-QT の結果が示されている. また, Rnd はランダムな行動をとる 2 体のハンタの結果であり, 比較のために示されている. 縦軸は所要ステップ数, 横軸はエピソード数である. 各グラフのプロットは, そのエピソード数までの平均所要ステップ数を 10 回の実験で平均したものである. たとえば 10^2 のプロットは, 各実験において 10^1+1 エピソードから 10^2 エピソードまでの平均所要ステップ数を求め, それを 10 回の実験で平均したものである.

6.5 議　　論

図 7 の縦軸の所要ステップ数は少ないほど良い. XCS は Rnd よりは良い性能であるものの, XCS-QT よりは劣る. これは 5 章までの議論を支持する結果である.

実際に一般化の差が性能の差になっていることを確認するため, ある実験での最終的な (状態 00 の) 分類子の例を**表 3** に示す. 状態が 00 のときは, ハンタにとって真上に他のハンタと獲物がいる状態である. ハンタ間で獲物を挟み撃ちにするならば, 上に行くという行動 0 をとるのが正しい. そして, 獲物が実際にハンタの間にいるときとそうでないときがあるので, 半分程度の割合で挟み撃ちに成功することから, 0.5 程度が予測値となるはずである.

表 3 獲得された分類子例 (追跡問題)
Table 3 Example of acquired classifiers (Hunter problem).

XCS				XCS-QT			
c	a	p	F	c	a	p	F
0 0	0	0.02	0.10	0 0	0	0.52	0.96
# #	0	0.02	0.99	# #	0	0.17	0.12

(ただし c, a, p, F はそれぞれ分類子の条件部, 行動部, 予測値, 適応度を表す)

表 3 の XCS-QT の分類子において, 条件部が 00 である分類子の予測値は 0.52 であり, 上記の議論と合致する. そしてこの分類子の適応度は高い. 一方で, 条件部が##で行動部が 0 の分類子は予測値は低く, 適応度も低い. これは条件部が##であるような分類子は不必要なので望ましい結果である.

ところで, 本実験では 5 章の木の問題とは異なり, 条件部には 1 つも#が必要ない問題であり, 一般化を議論するのにふさわしくない問題のように思われる. しかしながら, LCS においては分類子の#が増えるように圧がかかるため, #の入った不必要な分類子が高い適応度にならないことは, 一般化が正しく機能していることを示している.

表 3 の XCS の分類子において, 条件部が 00 である分類子の予測値は 0.02, 適応度は 0.10, 条件部が##である分類子の予測値は 0.02, 適応度は 0.99 である. これらの分類子が得られる理由は, 以下のようになる. 条件部が 00 で行動部が 0 の分類子は不可欠でありながらも, XCS では誤差値が大きくなってしまい, 条件部が##の分類子と同程度の適応度となる. この結果, ほとんど報酬が得られなくなる. そして, ほとんどの場合において予測値が小さい, 条件部が##の分類子の誤差値は小さくなる, という理由である.

これらの議論より, XCS が正しく一般化を実現できないようなマルチエージェント環境においても, XCS-QT がそれを克服し, 学習できることを示せたといえる.

7. 結　　論

本論文では, 適切な一般化に着眼したうえで, 決定的環境より複雑なマルチエージェント環境でも利用可能な XCS-QT を提案した. 実験では木の問題を用い, 決定的状態遷移環境, 確率的状態遷移環境, 部分知覚環境, マルチエージェント環境における性能の違いを検証した. 加えて, より複雑で一般的なマルチエージェント強化学習の枠組みである追跡問題を用いて性能の違いを検証した. これらの検証を通して, 経験の一般

化を正しく行う XCS-QT は，マルチエージェント環境において XCS よりも優れていることが分かった．

謝辞 本研究は情報通信研究機構の研究委託により実施したものである．本研究の一部は文部科学省の科学研究費補助金（基盤研究（B），課題番号 17360424）の支援によって行われた．

参 考 文 献

1) Holland, J.: Escaping Brittleness: the Possibilities of General-purpose Learning Algorithms Applied to parallel Rule-based Systems, *Machine Learning, An Artificial Intelligence Approach*, Mitchell, T., Michalski, R. and Carbonell, J. (Eds.), pp.593–623, Morgan Kaufmann (1986).
2) Sutton, R. and Barto, A.（著），三上貞芳，皆川雅章（訳）：強化学習，森北出版株式会社 (2000).
3) Wilson, S.: ZCS: A zeroth level classifier system, *Evolutionary Computation*, Vol.2, No.1, pp.1–18 (1994).
4) Wilson, S.: Classifier Fitness Based on Accuracy, *Evolutionary Computation*, Vol.3, No.2, pp.149–175 (1995).
5) 荒井幸代：マルチエージェント強化学習—実用化に向けての課題・理論・諸技術との融合，人工知能学会誌，Vol.16, No.4, pp.476–481 (2001).
6) Sen, S. and Sekaran, M.: Multiagent Coordination with Learning Classifier Systems, *Adaptation and Learning in Multi-agent systems*, Weiss, G. and Sen, S. (Eds.), pp.218–233, Springer-Verlag (1995).
7) Bull, L.: On ZCS in Multi-agent Environments, *Parallel Problem Solving from Nature — PPSN V*, Eiben, A., Baeck, T., Shoenauer, M. and Schwefel, H. (Eds.), pp.471–480, Springer-Verlag (1998).
8) Hercog, L. and Fogarty, T.: Co-evolutionary Classifier Systems for Multi-agent Simulation, *Proc. Congress of Evolutionary Computation 2002*, pp.1789–1803 (2002).
9) Goldberg, E.: *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addisonwesley (1989).
10) Bellman, R.: *Dynamic Programming*, Princeton University Press (1957).
11) Wilson, S.: Generalization in the XCS Classifier System, *Proc. Genetic Programming 1998*, pp.665–675 (1998).
12) Butz, M. and Wilson, S.: An Algorithmic Description of XCS, *Soft Computing*, Vol.6, pp.144–153 (2002).
13) Wilson, S.: Get Real! XCS with Continuous Valued Inputs, *Computer Science*, Vol.1813, pp.209–222 (2000).
14) Gasser, L., Rouquette, N., Hill, R. and Lieb, J.: Representing and Using Organizational Knowledge in Distributed AI Systems, *Distributed AI Systems*, Vol.2, pp.55–78, Morgan Kaufmann (1989).

(平成 17 年 10 月 4 日受付)
(平成 18 年 3 月 2 日採録)

井上 寛康

2000 年京都大学大学院情報学研究科修士課程修了．2000 年から 2002 年（株）日立製作所にてソフトウェア開発に従事．2002 年京都大学大学院情報学研究科博士後期課程に復学．同年より ATR にて研修研究員．2005 年京都大学大学院情報学研究科博士後期課程研究指導認定退学．同年より ATR にて研究員．マルチエージェント，強化学習の研究に従事．人工知能学会会員．

高玉 圭樹（正会員）

1998 年東京大学大学院工学系研究科博士課程修了．同年国際電気通信基礎技術研究所（ATR）入所．2002 年東京工業大学大学院総合理工学研究科講師，現在に至る．博士（工学）．マルチエージェントシステム，分散人工知能，強化学習，創発的計算手法の研究に従事．著書に『マルチエージェント学習—相互作用の謎に迫る』等．

下原 勝憲（正会員）

1978 年九州大学大学院工学研究科修士課程修了．同年電信電話公社横須賀電気通信研究所入所．1993 年 ATR 人間情報通信研究所第六研究室長，1999 年 NTT コミュニケーション科学基礎研究所社会情報研究部長を経て，2001 年 ATR 人間情報科学研究所所長．現在，京都大学大学院情報学研究科客員教授を兼任．博士（工学）．コミュニケーション創発機構の研究に従事．