電気通信大学大学院情報理工学研究科

平成 25 年度修士論文

不完全知覚環境のための複素強化学習に基づく学習分類子システム

学籍番号　1230078
氏名　山﨑 大地
総合情報学専攻　メディア情報学コース
主任指導教員　高玉 圭樹教授　印
指導教員　吉浦 裕教授　印

提出日　平成 26 年 1 月 30 日　(木)

概要

本研究では，難解な POMDPs 環境において最適な方策を獲得するために，複素強化学習に基づく学習分類子システムとして Complex-Valued Classifier System（CVCS）およびその改良である Adjustment-Population-size-based CVCS（AP-CVCS）を提案する．提案システムを用いた手法は，POMDPs 環境に適用可能な強化学習手法である複素強化学習（Complex-Valued Reinforcement Learning：CVRL）と比較して，1）方策（分類子）を進化計算より探索し，不要な分類子を淘汰することで，最適な方策を効率よく探索可能である．また，POMDPs 環境に適用可能な学習分類子システム（Learning Classifier System ：LCS）である進化型メモリベース法を組み込んだ ZCSM（Zeroth level Classifier System with Memory）と比較して，2)方策に内部メモリを用いず，行動履歴のみを用いることで，少ない計算リソースで POMDPs 環境における不完全知覚問題を解決することが可能である．

提案手法の有効性を検証するための計算機実験として，a）標準的な不完全知覚環境，およびに，従来手法が適用困難な POMDPs 環境として b）状態空間が大きい環境，c）不完全知覚の特性が異なる環境に提案手法を適用したところ，次の知見を得た．まず，1）提案手法は，従来手法（$\dot{Q}$ -learning と ZCSM）よりも少ない学習回数で高い学習性能を実現し，2）不完全知覚に対して必要なパラメータについて適切な値が設定できないために従来手法では学習不可能な環境においても学習が可能であり，3）従来手法が最適な方策を獲得不可能な初期状態が不完全知覚となる問題においても，提案手法は最適な方策を獲得可能であることを明らかにした．また， AP-CVCS に関しては，1) CVCS や従来手法と比較してより安定した最適方策の獲得を達成できる一方，2）初期状態が不完全知覚状態となるような環境では従来手法と同等に安定した性能を発揮することを確認した．

また，さらなる展開として，実問題への適用可能性について調査するため， 1) 不完全知覚に対して必要なパラメータについて環境に合わせて適用的に変化させる機構の考案および評価実験を行った結果，事前にパラメータを設定することなく，不完全知覚問題を解決することができた．また， 2）知覚入力に外乱の発生する環境下での評価実験を行った．その結果として，$\dot{Q}$ -learning と比較して CVCS の枠組がノイズに対して頑強性があることを示す一方，AP-CVCS では安定した学習が困難となることを明らかにした．

目次

第 1 章　序論 1

第 2 章　部分マルコフ決定過程 3

2.1　エージェントの状態知覚 3

2.2　不完全知覚問題の分類 4

2.2.1　Type 1 の不完全知覚 4

2.2.2　Type 2 の不完全知覚 5

2.2.3　Type 3 の不完全知覚 6

第 3 章　複素強化学習 7

3.1　強化学習 7

3.2　Q-learning 7

3.3　$\dot{Q}$ -learning 8

3.3.1　複素行動価値の更新 9

3.3.2　複素行動価値による行動選択 10

3.4　$\dot{Q}$ –learning の問題点 12

第 4 章　ZCSM (Zeroth level Classifier System with Memory) 14

4.1　ZCS の概要 14

4.2　分類子 15

4.3　メカニズム 15

4.3.1　実行部 15

4.3.2　強化部 16

4.3.3　発見部 16

4.4　ZCS のアルゴリズム 17

4.5　ZCSM への改良 17

4.6 ZCSM のアルゴリズム 18
4.7 ZCSM の問題点 19
第 5 章 CVCS (Complex-Valued Classifier System) 20
5.1 CVCS の位置付け 20
5.2 分類子 21
5.3 メカニズム 22
5.3.1 実行部 22
5.3.2 強化部 23
5.3.3 発見部 23
5.4 CVCS のアルゴリズム 24
5.5 AP-CVCS 24
5.5.1 分類子上限数の調整機構 26
5.5.2 選択圧保護機構 26
5.6 AP-CVCS のアルゴリズム 27
第 6 章 評価問題 29
6.1 Woods 問題 29
6.1.1 標準的な POMDPs 環境 30
6.1.2 広大な POMDPs 環境 30
6.1.3 Type 1 と Type 2 の混同がある POMDPs 環境 31
第 7 章 実験 1 CVCS の性能評価 33
7.1 評価基準とパラメータ設定 33
7.2 標準的な POMDPs 環境 36
7.3 広大な POMDPs 環境 38
7.4 Type 1 と Type 2 の混同がある POMDPs 環境 41
7.5 考察 44

7.5.1 提案手法と $\dot{Q}$ –Learning の比較 44
7.5.2 提案手法と ZCSM の比較 46
7.5.3 POMDPs の Type による提案手法の特性 47
7.5.4 複雑な POMDPs 環境における提案手法の特性 48
第 8 章 実験 2 AP-CVCS の性能評価 51
8.1 広大な POMDPs 環境 51
8.2 Type1 と Type2 が混在する POMDPs 環境 54
8.3 考察 58
第 9 章 さらなる展開 62
9.1 動的な基本位相の設定法 62
9.1.1 基本位相設定機構 62
9.1.2 評価実験 63
9.1.3 考察 64
9.2 知覚入力に外乱の発生する環境下での学習 65
9.2.1 評価実験 66
9.2.1 考察 68
第 10 章 結論 70
10.1 まとめ 70
10.2 今後の課題 71
参考文献 72
謝辞 74

第 1 章　序論

環境において目標とする状態に到達するために，適切な状態-行動制御則（方策）の獲得を目的として，強化学習（Reinforcement learning：RL）[18]と呼ばれる手法について研究が多くなされている．強化学習では，学習エージェントが知覚した環境内における自身の状態に応じて行動を実行することで，得られる報酬を最大化するような方策を獲得する．このような強化学習分野では，現在の状態と行動から次状態が確率的に決定するマルコフ決定過程（Markov Decision Processes：MDPs）を扱う環境を対象としてきた[13][16]．しかし，実環境では，環境内で発生する雑音（環境に含まれる外乱やエージェントの状態知覚誤差）により，異なる環境状態を同一状態と誤って知覚する不完全知覚問題が生じるため，MDPs 環境を想定することが困難である．そのため，現在の強化学習分野では，不完全知覚問題を有する部分観測マルコフ決定過程（Partially Observable Markov Decision Processes：POMDPs）[9]を扱う環境に適応することが重要な課題である [10][14]．POMDPs 環境に適用可能な RL 手法としては，決定的方策を用いずに確率的傾斜法によって確率的方策を改善する手法[20]やモンテカルロ法による政策評価と山登り法による政策改善を組み合わせた手法[7]などが提案されており，POMDPs 環境となる実環境の問題において強化学習法を適用した例も複数存在する[8][11]．その中でも近年，複素強化学習（Complex-Valued Reinforcement Learning：CVRL）[4]が注目されている．CVRL では強化学習の価値関数に複素数を用い，その価値を複素平面上で回転させることで行動文脈を構築し，行動履歴から不完全知覚状態を特定する．しかしながら，CVRL は学習器と環境の試行錯誤的なやり取りのみで学習を進めるため，複雑な POMDPs 環境のような報酬獲得に多数の試行が必要となる環境における学習効率が低下するという問題が存在する．

一方で，適切な方策を進化的に獲得する学習分類子システム（Learning Classifier System ： LCS）[5]が，近年盛んに強化学習問題に適用されている[6][22]．LCS は強化学習と遺伝的アルゴリズム（Genetic Algorithm：GA）を組み合わせたシステムであり，強化学習における方策を IF-THEN ルール（分類子）を用いて表現し，GA を用いて分類子を進化させることで，適切な方策を獲得する点が特徴である．したがって，全ての状態行動ルールを網羅的に探索することで学習に時間を必要とするという CVRL の問題に対し，LCS は複雑な MDPs 環境においても進化計算によって適切な方策を獲得し，不要なルールを除外することで効率的な学習が可能である．POMDPs 環境に適用可能な LCS として，進化型メモリベース法を組み込んだ ZCSM（Zeroth level Classifier

System with Memory）[2]が主流である．進化型メモリベース法の特徴は，蓄積された過去の状態行動履歴を記憶し，それに対応する履歴情報をメモリとして分類子に付加することで，不完全知覚状態を知覚するとともに，分類子および分類子に対する適切な履歴情報を進化的に探索することである．しかし，進化型メモリベース法の問題は，1）複数の不完全知覚状態を有する環境ではメモリサイズが膨大となること，加えて 2）メモリサイズの増加により，メモリと分類子の組み合わせ数が膨大となることで，進化計算法の探索効率が低下することである．したがって，ZCSM は，複雑な POMDPs 環境では最適な方策を獲得することが困難となる．

そこで本研究では，CVRL を学習分類子システムに組み込んだ Complex-Valued Classifier System（CVCS）およびその改良である Adjustment Population size based CVCS（AP-CVCS）を提案する．提案手法は，CVRL と比較して，1）進化計算を用いて謝った方策（分類子）を削除することで，適切な方策のみを短時間で効率的に学習可能である．また，ZCSM と比較して，2）不完全知覚状態の特定に複素行動価値による行動履歴を用いることで，メモリベース手法を用いる必要がなく，状態行動空間を増加させずに学習できる．そのため提案手法は，従来手法（$\dot{\text{Q}}$-Learning と ZCSM）が適用困難な POMDPs 環境においても，最適な方策を学習することが可能である．本研究では，提案手法の有効性を検証するため，仮想迷路環境（Woods 問題）での計算機実験から従来手法と学習性能を比較する．

以下，第 2 章では POMDPs 環境の特徴について述べ，第 3 章および第 4 章では CVRL および ZCSM について説明する．第 5 章では提案手法およびその改良について説明する．第 6 章では計算機実験で用いる評価問題について述べ，第 7 章および第 8 章で提案手法およびその改良に対する実験結果を示し考察する．第 9 章では実環境問題への適用のためのさらなる展開について述べ，最後に，第 10 章で本研究の結論および今後の課題についてまとめる．

第 2 章　部分マルコフ決定過程

POMDPs 環境下では MDPs 環境と異なり, エージェントの知覚入力が実際の状態に対して確率的に決定される．このような環境下では，学習エージェントは環境に対して自身の状態を正確に知覚することができない．そのため，状態に対する適切な行動を決定することが困難となる．また，POMDPs 環境の特徴から，本来は異なる方策を獲得すべき別々の状態を同一の状態として知覚する問題が生じることで，適切な方策の獲得が困難となる問題も存在する．このような問題は不完全知覚問題と呼ばれる[1]. Zhanna らは POMDPs 環境下においてモデル化される不完全知覚問題が発生する環境の分類について，報酬までのステップ数 d および最短ステップで報酬を獲得可能な行動 a の 2 つの要素が重要となると仮定した[23]. その上で, 不完全知覚となる 2 状態に対する 2 要素 (d_1, d_2, a_1, a_2)の関連性から, POMDPs 環境によって発生する不完全知覚問題を 3 種類のタイプに分類した．また，宮崎らは不完全知覚によって本来エージェントがおかれている状態を誤認してしまう状況を混同と定義し，Type 1 と Type 2 に関する混同を定義している[10]．以下，エージェントの状態知覚および Zhanna の定義した不完全知覚問題に関する 3 種類のタイプについて説明する．

2.1 エージェントの状態知覚

学習エージェントは，環境中において自身がおかれている状態sを知覚し，状態に応じて行動aを行うことで，自身の目的を達成するための方策を獲得する．状態に対する行動決定策（方策）は，状態と行動の組からなる行動価値$v(s, a)$によって決定される．例えば図 1 に示すグリッド上の環境中で，エージェントが自身の八近傍の状況から状態を知覚する場合，青枠で囲まれた地点（*）のどちらにエージェントが存在していても，エージェントは自身の状態を同一であると知覚してしまう．しかし，薄緑枠で囲まれた地点（**）は八近傍の状況が他の全てのマスと異なるため，他の状態とは異なる状態であると知覚することができる．

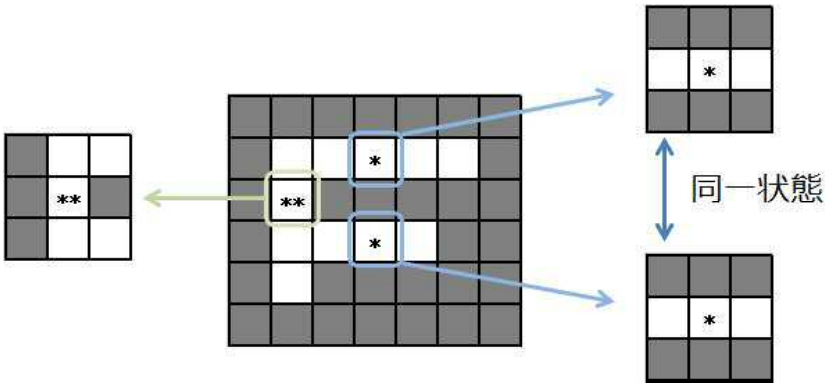

図 1　エージェントの状態知覚の例

2.2 不完全知覚問題の分類

2.2.1　Type 1 の不完全知覚

不完全知覚となる2状態に対する2要素が, $d_1 \neq d_2$かつ $a_1 = a_2$となる場合, その環境は Type 1 に分類される．また，Type1 の混同は，適切な行動（報酬に近づく行動）の行動価値が異なる状態を同一の状態として知覚することと定義されている．例えば，図 2 に示す環境のように，学習エージェントは状態 1a と 1b を同一の状態として知覚するが，各状態における行動価値を「10－報酬までの最短ステップ数」とした場合，状態 1a から 2 に遷移するため右に移動する行動の価値は$v(1a, \rightarrow) = 8$となり，状態 1b から状態 3 に遷移するため右に移動する行動価値は$v(1b, \rightarrow) = 2$となる．そのため，状態 1a と 1b に等確率で到達する場合，学習エージェントは状態 1 から右に移動する行動価値を$v(1, \rightarrow) = 5$と推定する．同様に，状態 3 から状態 4 に遷移するために右に移動する行動価値は$v(3, \rightarrow) = 3$となるが，価値を 5 と推定している状態 1 に移動するため状態 3 から左に移動する行動価値は$v(3, \leftarrow) = 4$と推定される．このように，報酬から遠ざかる行動価値を誤って高く見積もることで，適切な方策を獲得できなくなるという問題が生じる．

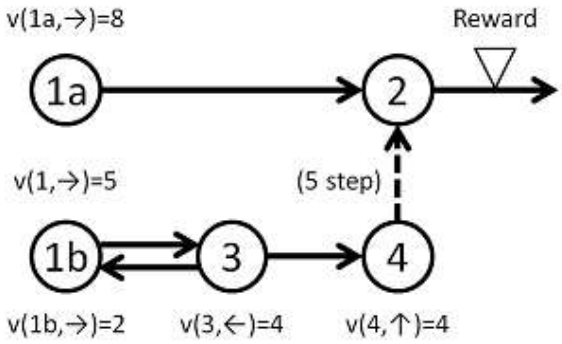

図 2　Type 1 の不完全知覚の例

2.2.2　Type 2 の不完全知覚

不完全知覚となる 2 状態に対する 2 要素が, $d_1 \neq d_2$かつ $a_1 \neq a_2$となる場合, その環境は Type 2 に分類される．Type 2 の混同は，合理的ルール（正の報酬を獲得可能な方策の構成要素となるルール）と非合理的ルール（合理的ルールでないルール）を同一ルールとして知覚することである．例えば，図 3 に示す環境のように，学習エージェントは状態 1a と 1b を同一の状態として知覚する．ここで初期状態を状態 S としたとき, 最短ステップで報酬を獲得するためには, 合理的ルールである状態 1a で右に移動する行動を学習する必要がある．ところが，エージェントが同じ状態と知覚した状態 1b では，状態 1b に遷移し続ける非合理的ルールとなるために報酬を獲得不能となる．同様に，状態 1b では上に移動する行動が合理的ルールであるが，同ルールを状態 1a に適用した場合，状態 1a と状態 2 を往復する非合理的ルールとなり，報酬を獲得できなくなる．このように，合理的ルールと非合理的ルールを同一ルールとして知覚することで, 適切な方策を獲得できなくなる問題が生じる．

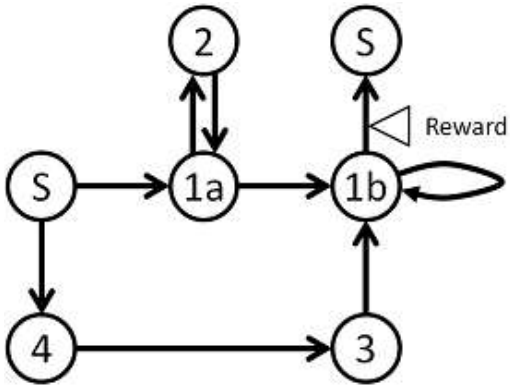

図 3　Type 2 の不完全知覚の例

2.2.3　Type 3 の不完全知覚

不完全知覚となる2状態に対する2要素が, $d_1 = d_2$かつ $a_1 \neq a_2$となる場合, その環境は Type 3 に分類される. Type 3 の環境では不完全知覚となる 2 状態の報酬までの距離が同じであるため，初期状態から報酬までに 1 つの不完全状態しか観測しない．そのため，初期状態に応じて不完全知覚状態における適切な行動が変化することとなる．例えば，図 8 に示す環境のように，学習エージェントは状態 1a と 1b を同一の状態として知覚され，初期状態は状態 Sa または Sb からランダムに決定される．ここで初期状態が Sa となった場合，最短ステップで報酬を獲得するためには，合理的ルールである状態 1a で右に移動する行動を学習する必要がある．ところが，エージェントが同じ状態と知覚した状態 1b では，状態 Sb と状態 1b を往復する非合理的ルールとなるために報酬を獲得不能となる．同様に，初期状態が Sb となった場合，状態 1b では左に移動する行動が合理的ルールであるが，同ルールを状態 1a に適用した場合，状態 1a と状態 Sb を往復する非合理的ルールとなり，報酬を獲得できなくなる．

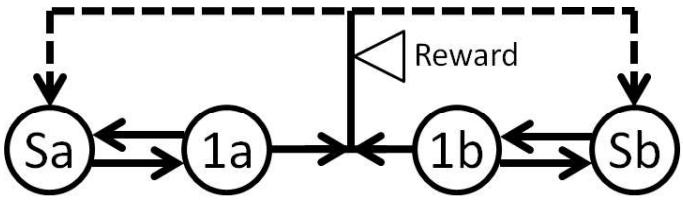

図 4　Type 3 の不完全知覚の例

第 3 章　複素強化学習

本章では，複素強化学習が用いる強化学習およびその一手法である Q-learning[19]について説明する．その後，複素強化学習法に改良した $\dot{\text{Q}}$-learning について述べる．

3.1 強化学習

強化学習とは，試行錯誤的な行動を通して，エージェントを環境に適応させる適切な振舞を獲得させることを目的とする機械学習法の一種である．強化学習の概念図を図 5 に示す．強化学習エージェントは，ある時刻tにおいて環境から受け取った自身の状態$\mathrm{s_t}$にもとづき行動$\mathrm{a_t}$を選択する．また，エージェントは選択した行動を実行した結果，環境から報酬$\mathrm{r_t}$を受け取る．エージェントはこの一連の処理を繰り返しながら，より多くの報酬を獲得可能な行動選択の指針（方策）を学習する．強化学習は他の機械学習法と異なり，教師信号を必要とせず，エージェントと環境の相互作用によって目標達成を目指す．そのため，エージェントは環境に対する予備知識を必要とせず，未知の環境や不確実性を含む環境においても学習を進めることができるという特徴を有する．

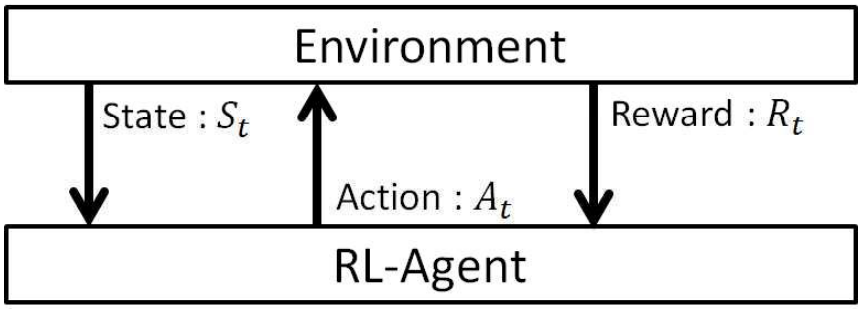

図 5　強化学習の概念図

3.2 Q-learning

Q-learning では，エージェントが環境から入力された状態sと行動価値関数$Q(s,a)$から，ルーレット選択やボルツマン選択によって行動aを選択し，得られた報酬を基に適切な方策を学習する．具体的には，環境から与えられた報酬r_{t+1}を用いて，式(1)により$Q(s,a)$の更新を行う．パラメータαおよびγは学習率および割引率であり，それぞれ学習結果が与える影響の強さ，将来の報酬を考慮する割合を制御するパラメータである．

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha\left(r_{t+1} + \gamma \max_a Q(s_{t+1}, a) - Q(s_t, a_t)\right) \quad (1)$$

Q-learning は，MDPs 環境下において適切なパラメータ設定の元に無限回の試行を行った場合，最適な方策を獲得可能であることが保障されている[18].

3.3 $\dot{Q}$ -learning

POMDPs 環境下の Q-learning 手法として，価値関数を複素数化することで方策の文脈を表現可能にした $\dot{Q}$–learning[15]が提案されている.

$\dot{Q}$-learning における学習エージェントは，複素数で表現される複素行動価値 $\dot{Q}(s, a)$を持つ．その価値の大きさは絶対値によって表され，時系列上での文脈情報は位相によって表される．また，学習エージェントは複素数である内部参照値 $\dot{\imath}$ を保持し，その位相は 1 ステップ前の複素行動価値から基本位相 $\dot{\beta}$ だけ逆回転した値となる．$\dot{Q}$-learning ではこの内部参照値に近い位相を持つ複素行動価値を持つ行動を優先的に選択することで，時系列に対する方策の文脈を表現することができる．行動選択法の詳細については後述する．ここで，複素行動価値の表現例を図 6 に示す．図中の右上にある複素行動価値$\dot{Q}(1, \uparrow)$は絶対値が 2，位相が$\pi \cdot 1/3$となる.

例として, 図 7 に示す環境において$\arg \dot{\beta} = \pi \cdot 1/3$と設定されている場合を考える．初期状態 S において図 6 にある位相$\pi \cdot 3/3$を持つ行動が実行された結果として状態 1 に遷移した場合，次のステップの内部参照値 $\dot{\imath}$ は$\pi \cdot 2/3$となるため, 位相が$\pi \cdot 1/3$である$\dot{Q}(1, \uparrow)$よりも位相が$\pi \cdot 2/3$に近い行動価値$\dot{Q}(1, \rightarrow)$が選択される．その次のステップではさらに内部参照値が回転して$\pi \cdot 1/3$となるため，$\dot{Q}(1, \uparrow)$が選択される．このようにして，位相によって時系列の文脈（行動の選択順）を表現する.

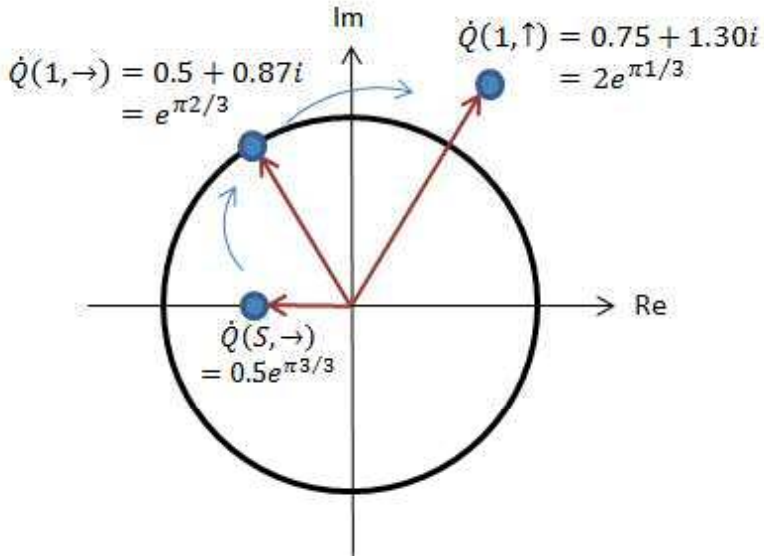

図 6　時系列に対する方策の文脈

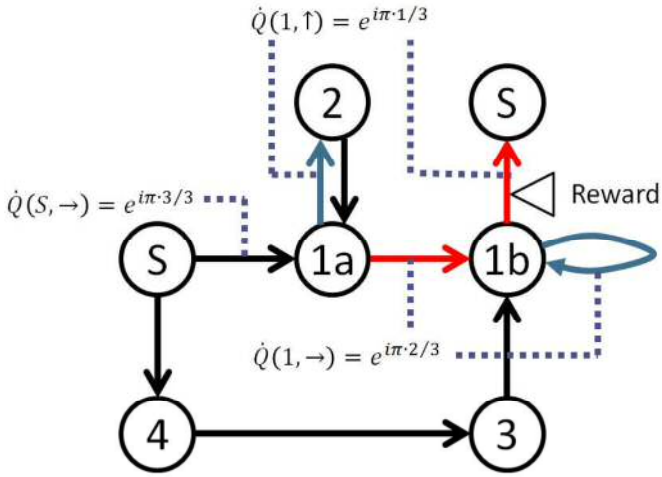

図 7　不完全知覚における方策の位相の具体例

3.3.1　複素行動価値の更新

$\dot{\text{Q}}$-learning において環境から与えられた報酬に対する行動価値関数の更新式は，以下の式(2)，式(3)で表される．

$$\dot{Q}(s_t, a_t) \leftarrow (1-\alpha)\dot{Q}(s_t, a_t) + \alpha\left(r_{t+1} + \gamma \dot{Q}^{(t)}{}_{max}\right)\dot{\beta} \tag{2}$$

$$\dot{Q}^{(t)}{}_{max} = \dot{Q}\left(s_{t+1}, \underset{a}{\text{argmax}}\left(Re\left[\dot{Q}(s_t, a)\bar{\dot{I}}_t\right]\right)\right) \tag{3}$$

ここで $\dot{I}_t$ は時刻 t における内部参照値，$\bar{\dot{I}}$ は $\dot{I}$ の複素共役，$Re[\cdot]$は複素数の実部を表す．これによって$\dot{Q}^{(t)}{}_{max}$ には，次状態の行動価値関数の中でも内部

参照値に近く，かつ，絶対値の大きい行動価値が選択される．Q-learning における更新式と異なり，次状態での行動価値$\dot{Q}^{(t)}{}_{max}$から位相が$\dot{\beta}$ だけ回転した値に近づくように行動価値関数が更新される．

行動価値関数の位相によって時系列に対する方策の文脈が表現される様子を図 8 に示す．ここで学習率$\alpha = 0.2$，割引率$\gamma = 1.0$，基本位相$\dot{\beta} = \pi \cdot 1/3$，報酬値$r_{t+1} = 0$とした場合，更新後の$\dot{Q}(s_t, a_t)$は次ステップの複素行動価値$\dot{Q}^{(t)}{}_{max}$から$\dot{\beta}$だけ回転した位相に近づくことになる．最終的に複素行動価値は 1 ステップごとに$\dot{\beta}$ずつ逆回転するような位相を持つが，$\dot{Q}(s_t, a_t)$ および $\dot{Q}^{(t)}{}_{max}$の値によっては，行動価値の位相が基本位相の整数倍とならなくなる場合もある．

Q̇ -learning における行動価値関数の更新では, 過去の価値関数を反映した文脈の獲得や学習速度の向上を目的に適格度トレース[18]を適用している. 式(2)に適格度トレースを適用した場合の更新式は以下の式(4)のように表すことができる.

$$\dot{Q}(s_{t-k}, a_{t-k}) \leftarrow (1-\alpha)\dot{Q}(s_{t-k}, a_{t-k}) + \alpha\left(r_{t+1} + \gamma \dot{Q}^{(t)}{}_{max}\right)\dot{\beta}^{k+1} \qquad (4)$$

ここで $k = 0,1,\ldots,Ne-1$であり，Neは何ステップ前の価値関数を参照するか決定するトレース数と呼ばれるパラメータである．$Ne = 1$の場合，更新式は式(2)で表現できる．

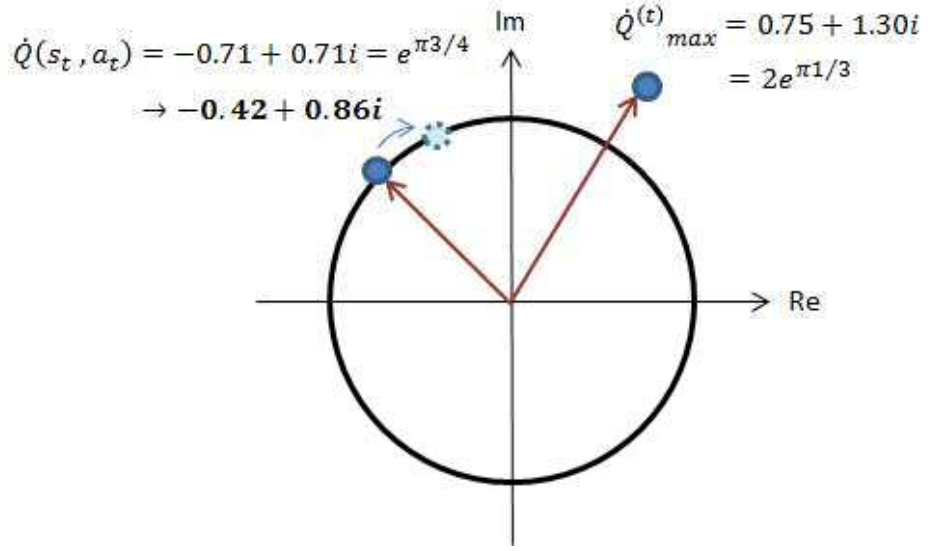

図 8　複素行動価値の更新の様子

3.3.2　複素行動価値による行動選択

Q̇ -learning における行動選択では，(1) 複素行動価値の絶対値の大きさ (2) 複素行動価値の位相と内部参照値の位相の近さという 2 つの観点から行動を決定する．絶対値の大きさは従来の強化学習と同様に，将来的な期待収益が大き

い行動を選択する指針となる．一方，内部参照値との位相の近さは時系列の文脈に沿った行動を選択する指針となる．

内部参照値は式(5)，式(6)で表される値を取る．ここで$\dot{I}_{-1}$は，初期状態における行動選択の際に参照される内部参照値である．内部参照値の位相は，直前に選択した複素行動価値に対して，式(3)によって回転した位相$\dot{\beta}$と逆の回転を加えたものとなる．式(5)によって更新された複素行動価値が収束した場合，次状態の行動価値の位相と内部参照値の位相が一致することから，このような内部参照値によって時系列の文脈が表現可能となる．なお，初期状態においては直前の複素行動価値が存在しないため，式(6)に示すように初期状態で最も絶対値の大きい複素行動価値を用いる．

$$\dot{I}_t = \dot{Q}(s_t, a_t)/\dot{\beta} \tag{5}$$

$$\dot{I}_{-1} = \dot{Q}\left(s_0, \underset{a}{\mathrm{argmax}}\left(\left|\dot{Q}(s_0, a)\right|\right)\right) \tag{6}$$

先述した行動選択に関する2つの観点の両方を考慮した行動選択方策として，式(7)に示すような複素行動価値のためのボルツマン選択法が提案されている．ここで$\pi_{i_{t-1}}(s_t, a')$は時刻tの状態s_tにおいて行動a'を選択する確率である．また，$T(>0)$は温度定数と呼ばれるパラメータである．式(7)中の$Re\left[\dot{Q}(s_t, a')\overline{\dot{I}_{t-1}}\right]$という表現は，式(3)において$\dot{Q}^{(t)}{}_{max}$を決定する際にも用いられているように，内部参照値に近く，かつ絶対値の大きい行動価値ほど大きな値となる．

$$\pi_{i_{t-1}}(s_t, a') = \frac{\exp\left(Re\left[\dot{Q}(s_t, a')\overline{\dot{I}_{t-1}}\right]/T\right)}{\sum_a \exp\left(Re\left[\dot{Q}(s_t, a)\overline{\dot{I}_{t-1}}\right]/T\right)} \tag{7}$$

この2つの条件を視覚的に表現した図を図9および図10に示す．式(7)中の$Re\left[\dot{Q}(s_t, a')\overline{\dot{I}_{t-1}}\right]$は，内部参照値から原点の直線に対して複素行動価値から垂線を引いた際の，その交点と原点の距離で表すことができる．図9において$\dot{Q}_1$と$\dot{Q}_2$は位相が等しく絶対値が異なる関係にあるが，図から分かる通り$Re\left[\dot{Q}(s_t, a')\overline{\dot{I}_{t-1}}\right]$の値は絶対値が大きい$\dot{Q}_1$の方が大きくなるため，絶対値が大きいほど行動選択されやすいことが分かる．また，図10において$\dot{Q}_1$と$\dot{Q}_2$は絶対値が等しく位相が異なる関係にあるが，同様に$Re\left[\dot{Q}(s_t, a')\overline{\dot{I}_{t-1}}\right]$の値は位相が内部参照値に近い$\dot{Q}_1$の方が大きくなるため，位相が内部参照値に近いほど行動選択されやすいことが分かる．

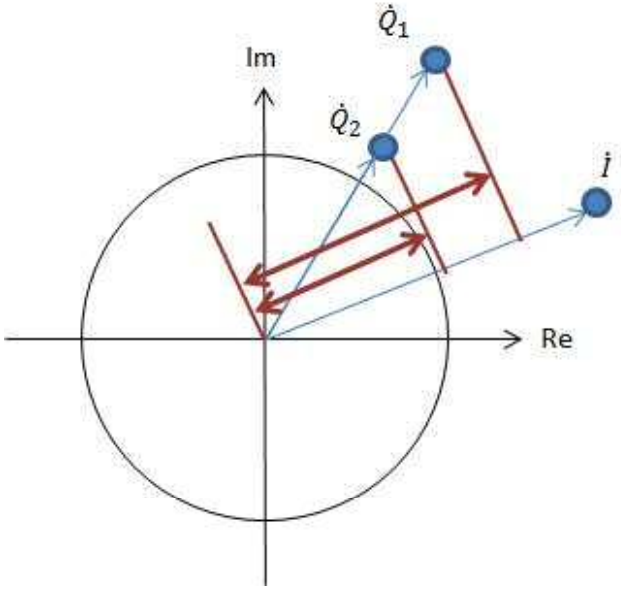

図 9　行動選択における複素行動価値の絶対値の影響

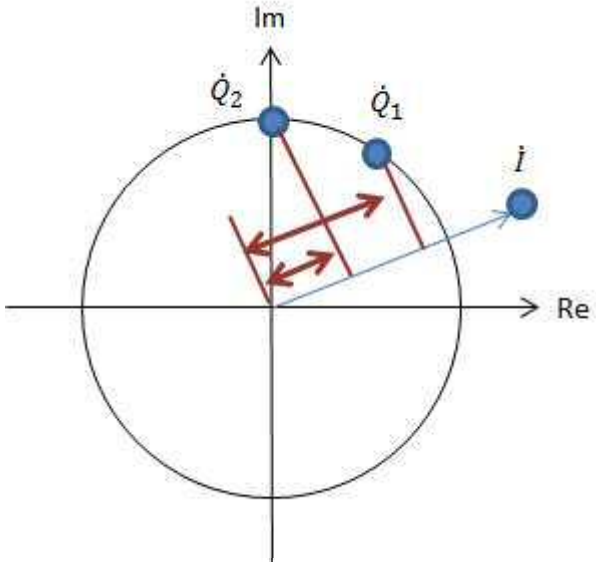

図 10　行動選択における複素行動価値の位相の影響

3.4 $\dot{Q}$ –learning の問題点

$\dot{Q}$–Learning では行動価値関数によって方策を表現するため, 適切な方策の獲得のためには全状態の数と全行動の数を掛け合わせた値と同じだけの行動価値を保持する必要がある．状態数や行動の種類が多くなる環境では計算に多くのメモリが必要となるため，環境の規模によっては学習が困難となる場合がある．また，エージェントが経験した状態および行動に関する価値関数のみを更新す

るために，報酬獲得機会が少ない環境においては学習効率が低下するという問題が存在するため，複雑な POMDPs 環境においても適用限界が存在する．

第 4 章 ZCSM (Zeroth level Classifier System with Memory)

POMDPs 環境に適用可能な LCS である ZCSM（Zeroth level Classifier System with Memory）について説明する．まず，ZCSM を構成する学習分類子システムである ZCS[21]について述べる．その後，ZCSM のメカニズムについて述べる．

4.1 ZCS の概要

ZCS は，学習と進化の 2 つの概念を取り入れた環境適応システムである．ZCS の概念図を図 11 に示す．ZCS は条件部と行動部からなる IF-THEN ルール（分類子）とそれらのルールから構成されるルール集合（Population : **[*P*]**）を持つ．また，ZCS は，環境状態を検出器（Detector）より状態を知覚する．さらに，ZCS の出力は効果器（Effector）を通じて環境に対する行動に変換される．ZCS におけるルールは，そのルールを実行したことで得られる報酬を予測した強度値をもつ．この強度値は，行動に対する見返りとして環境から獲得した報酬を用いて，RL により評価される．ここで，ZCS の目的は，強化学習と同様に，報酬を最大化するような方策を示すルールを学習することである．さらに，遺伝的アルゴリズム（Genetic Algorithm : GA）[3]を用いて分類子を進化させることで，膨大な状態行動空間から，最適なルールを探索する．

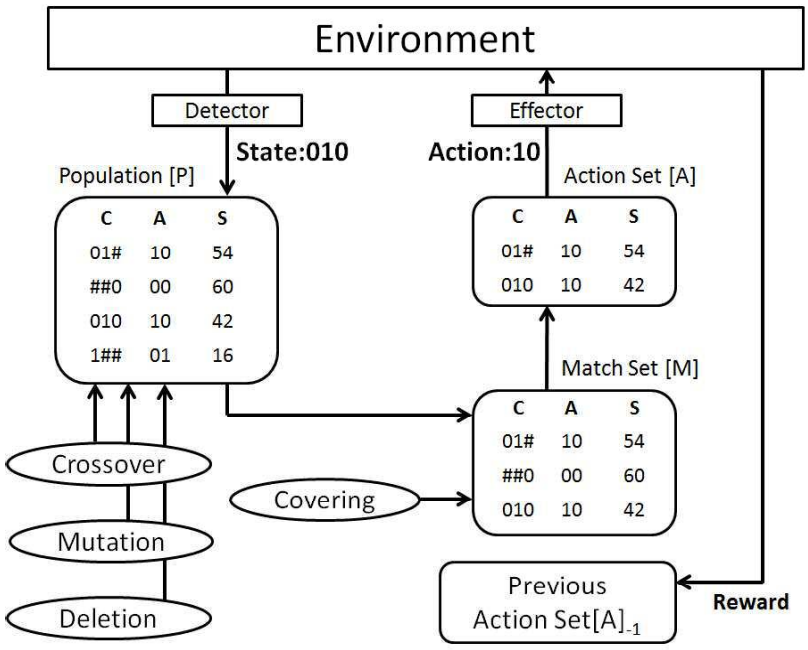

図 11　ZCS の概念図

4.2 分類子

ZCS における分類子は条件部（condition：C）と行動部（action：A），そのルールの価値である強度値（Strength：S）から構成される．条件部および行動部は 0,1 によって表現される．条件部は，エージェントが知覚した状態において，そのルールが使用可能であるか決定するために用いられる．そして，行動部はその分類子が使用される際の出力を意味する．ここで，条件部については任意の値を意味する#（don't care）記号を組み込むことで，複数の状態に使用可能な汎用的な分類子を生成できる．

4.3 メカニズム

ZCS のメカニズムは，（1）実行部，（2）強化部，（3）発見部から構成される．

4.3.1　実行部

実行部では，環境から入力された状態において，適切な行動を出力するまでの処理を行う．環境から入力された状態は 0,1 のビット列から形成される．ここで，条件部と入力状態が一致するルールを[*P*]から選択し，それらの分類子によって照合集合（Match Set：[M]）を形成する．ここで，1）入力状態に一致する分類子が存在しない場合，もしくは 2) [*M*]内の分類子の強度値の合計が，[*P*]内の分類子の強度値の平均にパラメータφを掛けた値よりも低い場合（[*M*]内に有

望な分類子が存在しない場合），入力状態に一致する条件部を持つ分類子を生成する．この処理を被覆（Covering）と呼ぶ．被覆によって生成される分類子の条件部の各ビットは，確率 $P_{\#}$で#に置き換えられ，行動部はランダムに設定される．Covering によって生成される分類子の強度値は初期値 S_0に設定する．

$[M]$を形成後，要素数 N である$[M]$内の分類子cl_iの強度値を適合度f_iとし，式(8)に示すルーレット選択による選択確率P_iから分類子を 1 つ選択し，実行する行動を選択する．ルーレット選択では，元となる集合から式(8)に示す確率によって要素を 1 つ選択する．そして，実行する行動を行動部にもつ分類子を$[M]$から選択し，行動集合（Action Set : [A]）を形成する．その後，行動を環境に対して実行する．この一連の処理の流れを 1 ステップと呼ぶ．

$$P_i = \frac{f_i}{\sum_{j=1}^{N} f_j} \tag{8}$$

4.3.2 強化部

強化部では，実行部を実行後，環境から得られた報酬に基づいて，1 ステップ前の行動集合$[A_{-1}]$内の分類子の強度値を更新する．具体的には$[A_{-1}]$内の分類子cl_iの強度S_iは，式(9)を用いて更新する．ここでrは環境からの報酬値，$[A']$は次ステップにおいて，分類子の強度値の合計が最大となる行動をもつ分類子から形成される行動集合，$|A_{-1}|$は 1 ステップ前の行動集合内の分類子数をそれぞれ意味する．パラメータ$\alpha(\epsilon[0,1])$および$\gamma(\epsilon[0,1])$は学習率，割引率である．

$$S_i \leftarrow S_i + \alpha\left(\left(r + \gamma \sum_{cl_j \in [A']} S_j\right) \Big/ |A_{-1}| - S_i\right) \tag{9}$$

また，1 ステップ前の照合集合内に含まれているが行動集合内に含まれていない各分類子については，その強度にパラメータτが掛けられることで強度が減衰する．

4.3.3 発見部

発見部では，GA を用いて$[P]$内の分類子を進化させることで，適切なルールを探索する．GA は，エージェントの学習が終了するたびパラメータρの確率で実行される．GA が実行される場合，$[P]$内の分類子の強度値を適合度とし，ルーレット選択による選択確率から親個体となる分類子が 2 つ選択される．次に，子個体として，各親個体の分類子と同様の条件部と行動部および強度値を持つ分類子を 2 つ生成し，交叉（crossover）および突然変異（mutation）を適用する．交叉はパラメータχの確率で実行され，2 つの子個体の条件部の一部を交換することで新たな条件部を持つ個体に進化させる．突然変異は，各個体の持つ条件部の各ビットについて，確率μでランダムなビットに変化させる．最後に，

生成した子個体を[*P*]に追加する．この際，[*P*]の分類子数が分類子上限数 N を超えた場合，分類子の強度値の逆数を適合度として，ルーレット選択による選択確率から削除する分類子を決定し削除する．

4.4 ZCS のアルゴリズム

前節で述べたメカニズムを備えた ZCS のアルゴリズムを図 12 に示す．学習エージェントは環境中から自身の状態を知覚し，その状態にマッチする分類子から[*M*]を生成する．その中から価値に応じて行動を選択し，[*A*]を生成して実行する．その結果として得た報酬や次状態の[*M*] の価値から[*A*]内の分類子の価値を更新する．これを報酬獲得（図中の end of problem）まで繰り返した後に，GA によって新たな分類子の探索を行う．この一連の流れを繰り返すことで，学習エージェントは最適な方策を獲得する．

```
while (! end of iteration)
  while (! end of problem)
    state ← recognizing an input from environment  ：環境から状態を知覚
    generate  [M]  ：[P]から state が照合する分類子を選択
    action  ← selecting the action from [M]
    generate  [A]  ：[M]から action, internal action が照合する分類子を選択
    reward  ←execute action：環境から報酬を獲得
    parameter update  ：[A], [M]の Strength 更新（強化部）
  end while
  run GA on [P]  ：[P] を対象に GA を実行（発見部）
end while
```

図 12　ZCS のアルゴリズム

4.5 ZCSM への改良

ZCSM[2]は，LCS を POMDPs 環境下に適用するために，メモリベース法として内部レジスタ(Inner Register)を ZCS に組み込んだシステムである. ZCSM の概念図を図 13 に示す.

ZCSM における学習エージェントは 0,1 のビット列から形成される b ビットの内部レジスタを持つ．内部レジスタは，学習開始時にすべて 0 で初期化される．ZCSM における分類子は条件部，行動部に加えて内部条件部（Internal Condition：IC），内部行動部（Internal Action：IA）が追加されている．内部条件部，内部行動部は内部レジスタと同様に長さ b のビット列であるが，通常

の条件部と同じく 0,1,#から構成される．

ZCSM の実行部では，環境入力と条件部が一致し，かつ，内部レジスタ（内部状態）と分類子の内部条件部が一致した分類子から[M]を形成する．行動選択では，環境に対して実行する行動に加えて，[M]内の分類子が持つ内部行動部から内部行動を選択する．その後，[M]から行動部と内部行動部がそれぞれ選択した行動と内部行動に一致する分類子を選択し，それらの分類子から[A]を形成する．環境に対して行動を出力後，内部行動は，内部レジスタを変更するために実行される．具体的には，内部レジスタの値は内部行動部の値で置き換えられるが，内部行動が #である場合は内部レジスタを変更しない．

ZCSM の発見部は，ZCS と同様であるが，分類子の内部条件部や内部行動部も交叉や突然変異が適用される．その結果，不完全知覚において適切な方策を示すような，内部条件部と内部行動部の適切な組み合わせをもつルールを進化的に獲得することが可能となる．

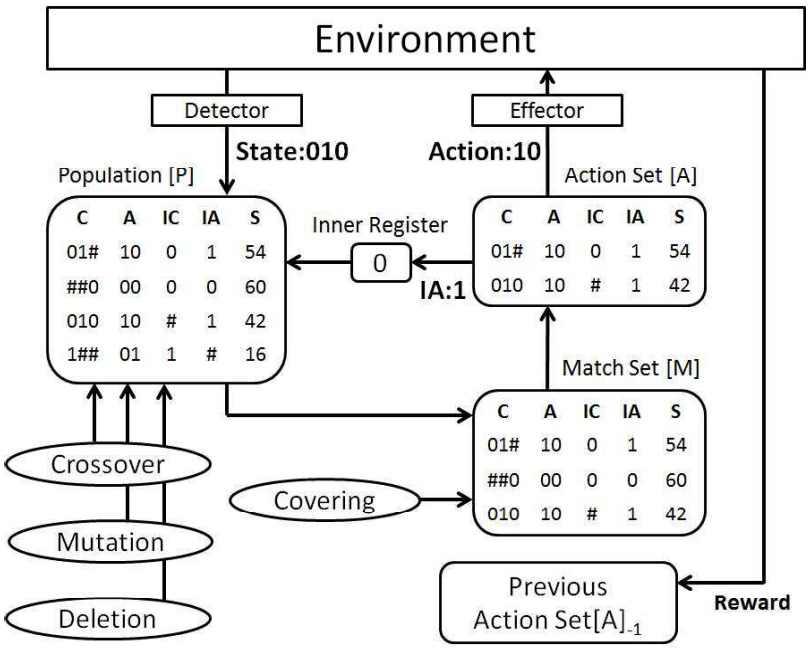

図 13　ZCSM の概念図

4.6 ZCSM のアルゴリズム

前節で述べたメカニズムを備えた ZCSM のアルゴリズムを図 14 に示す．4.4 節における図 12 との差異は，内部条件部および内部行動部に関する処理が追加された点である．学習エージェントは自身の状態と条件部，内部レジスタと内部条件部がそれぞれマッチする分類子から[M]を生成する．その中から価値に応じて行動および内部行動を選択し，[A]を生成して実行する．内部行動に従

って内部レジスタを更新し，行動の結果として得た報酬や次状態の**[*M*]** の価値から**[*A*]**内の分類子の価値を更新する．

```
while (! end of iteration)
   while (! end of problem)
      state ← recognizing an input from environment：環境から情報を知覚
      generate   [M]  ：[P]から state, internal state が照合する分類子を選択
      action, internal action   ← selecting the action from [M]
      generate   [A]  ：[M]から action, internal action が照合する分類子を選択
      reward   ← execute action and internal action：環境から報酬を獲得
      parameter update    ：[A], [M]の Strength 更新（強化部）
   end while
   run GA on [P]  ：[P] を対象に GA を実行（発見部）
end while
```

図 14　ZCSM のアルゴリズム

4.7 ZCSM の問題点

ZCSM において，内部レジスタのビット長は少なくとも不完全知覚状態の存在数を考慮して設定される．例えば，不完全知覚状態の組み合わせが 3 つであれば，2 ビット必要となる．しかしながら，ZCSM では内部レジスタ，内部条件部および内部行動部を扱うことで，学習エージェントが探索すべき空間は大きく増加する．内部レジスタが b ビットの場合，内部行動部および内部条件部によって状態行動空間がそれぞれ 2^b 倍ずつ増加する．内部レジスタを用いない分類子と比べて状態空間は 2^{2b} 倍に及ぶことから，状態空間が広く，複雑な不完全知覚問題を有する問題であるほど ZCSM の探索効率は大きく低下するという問題点がある．

第 5 章　CVCS（Complex-Valued Classifier System)

本章では，POMDPs 環境のための従来手法の問題点を克服し，より複雑な問題環境における適切な方策を獲得するため，$\dot{\mathrm{Q}}$–Learning を学習分類子システムに組み込んだ Complex-Valued Classifier System（CVCS）を提案する．以下，CVCS の位置づけおよび分類子による複素行動価値の表現について説明し，最後に CVCS のメカニズムについて述べる．

5.1 CVCS の位置付け

$\dot{\mathrm{Q}}$–Learning は全状態行動価値を保持する必要があるものの，メモリを使用せず行動価値$\dot{Q}(s,a)$と内部参照値$\dot{I}_t$のみから方策の文脈を表現することが可能である．しかし，全状態行動空間を網羅的に探索する必要があるため，広大な状態行動空間を有する環境においては学習効率が低下するという問題が存在するため，複雑な POMDPs 環境には適用限界が存在する．一方．ZCSM は，適切な方策を進化的に探索・学習することが可能であるため，全状態行動空間を網羅的に探索するために広大な状態行動空間を有する環境においては学習効率が低下するという $\dot{\mathrm{Q}}$–Learning の問題を克服できるメカニズムを有する．しかし，ZCSM はメモリベース法を用いるため，複雑な環境ではメモリ長の増加に伴って状態行動空間が爆発的に増加することで，探索効率が低下し，不完全知覚状態が特定困難となる問題が存在する．

提案手法である CVCS は，$\dot{\mathrm{Q}}$–Learning の問題点を，1）方策（分類子）を進化計算より探索し，探索効率を向上させることで克服する．さらに，ZCSM の問題点に対し，2）$\dot{\mathrm{Q}}$–Learning を扱い文脈依存型学習を行うことで，メモリを必要とせず不完全知覚状態を知覚可能である．これらを実現するための工夫として，CVCS では 1）時系列の文脈の崩壊につながるため，複数の状態で一致するような分類子の生成を防ぐ目的から，ZCSM で用いられている Don't care 記号を用いない．そのため，進化計算という観点からは淘汰の処理のみを実行する．また，2）同じく時系列の文脈の崩壊につながるため，行動順を表現する分類子の強度の位相を保護する目的から，ZCSM で用いられている交叉や突然変異の処理を行わない．さらに，3）$\dot{\mathrm{Q}}$–Learning における内部参照値 $\dot{I}$ から適切な価値を見積もって淘汰の対象を決定するため，$[A]$の要素数が 3 以上の場合には$[A]$から個体の淘汰を行う．ここで要素数が 3 以上の場合と設定されている

理由は，親個体として選択された強度値の高い個体がそのまま削除されてしまうことを防ぐためである．[A]の要素数が少ない場合には図 15 に示すように選択圧の変動を防ぎ適切な分類子を保護する目的から，[P]から淘汰を行う．これに伴って，この生成は常に[A]から実行される．以上の枠組を実装した CVCS の概念図を図 16 に示す．

図 15　CVCS における淘汰対象の決定

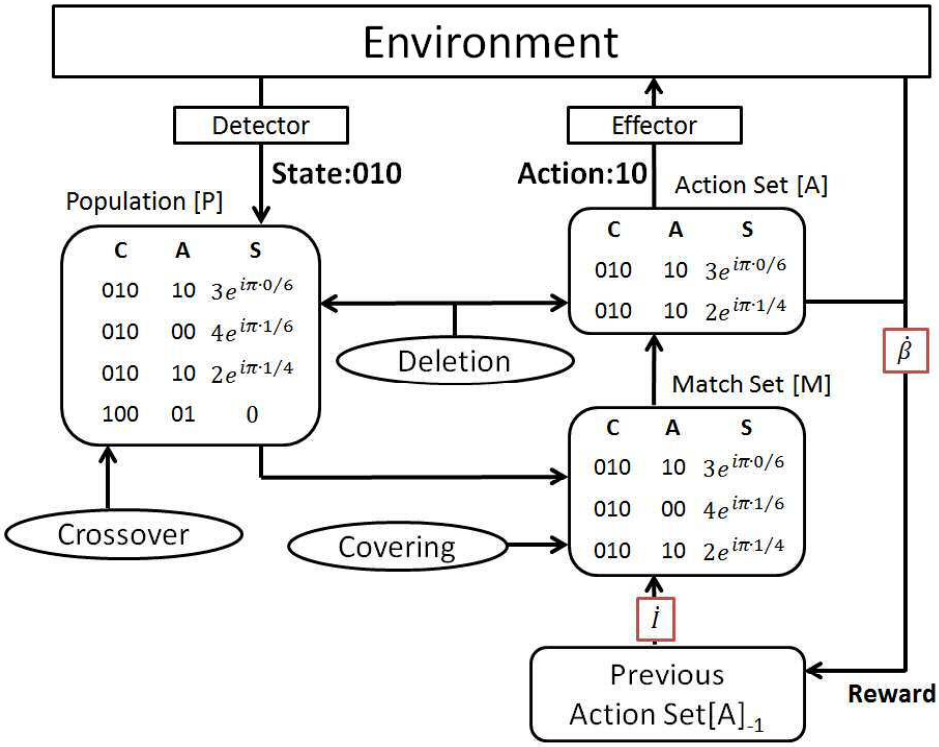

図 16　CVCS の概念図

5.2 分類子

CVCS で扱う分類子は，ZCS と同様に条件部ならびに行動部から構成される

が，強度値の代わりに Q̇–Learning における複素行動価値$\dot{Q}(s,a)$を用いる．したがって，ZCSM で必要な内部条件部ならびに内部行動部を用いる必要がない．しかし，#を条件部に含む汎用的な分類子は誤った文脈を形成するため，CVCS における分類子は # を用いない．同様に，誤った文脈の形成を防ぐため，$\dot{Q}(s,a)$の初期値は 0 とする．

また，Q̇–Learning の内部参照値とは異なり，CVCS における内部参照値は式(10)，式(11)により示される．ここで$[\boldsymbol{A_t}]$は時刻tにおける**[A]**内の分類子の集合，$[\boldsymbol{M_0}]$は初期状態（$t=0$）における**[M]**内の分類子の集合を表す．**[M]**内には様々な行動部を持つ分類子が存在する可能性があり，強度値の位相も様々な値をとることが考えられる．そのような状況下において強度値の合計を評価すると適切な位相を得られない場合がある．そのため，式(10)では$\dot{I}_t$を**[A]**内の分類子の強度の合計で評価する一方，式(11)では$\dot{I}_{-1}$を**[M]**内の単一の分類子の強度によって評価している．

$$\dot{I}_t = \sum_{cl_i \in [A_t]} \dot{S}_i / \dot{\beta} \tag{10}$$

$$\dot{I}_{-1} = \underset{cl_i \in [M_0]}{\operatorname{argmax}}\left(\left|\dot{S}_i\right|\right) \tag{11}$$

5.3 メカニズム

5.3.1 実行部

CVCS は ZCS と同様に，環境からの入力に対して**[P]**から条件部が一致する分類子を抽出し，**[M]**を形成する．その後，環境に出力する行動を，式(12)の確率に従って選択する．ただし，式中の$[\boldsymbol{M}]|a$は，**[M]**内で行動部がaである分類子の集合を表す．また，**[M]**内で行動部がaである分類子が存在しない場合，式(11)中の$\sum_{cl_i \in [M]|a} Re\left[\dot{S}_i \overline{\dot{I}_{t-1}}\right]$は 0 として扱う．

$$\pi_{\dot{I}_{t-1}}(a') = \frac{\exp\left(\sum_{cl_i \in [M]|a'} Re\left[\dot{S}_i \overline{\dot{I}_{t-1}}\right]/T\right)}{\sum_a \exp\left(\sum_{cl_j \in [M]|a} Re\left[\dot{S}_j \overline{\dot{I}_{t-1}}\right]/T\right)} \tag{12}$$

Covering の処理は，入力と条件部が一致する分類子が分類子集合内に存在しない場合，もしくは選択された行動と一致する行動部を持つ分類子が照合集合内に存在しない場合に実行する．後者の条件によって Covering が実行された場合，Covering によって生成される分類子の行動部は選択された行動と同一とな

る. また, Covering によって生成された分類子は時系列の文脈を持たないよう, 強度値は 0 に設定される.

5.3.2 強化部

環境から報酬rについて, kステップ前までの行動集合$[\boldsymbol{A_{t-k}}]$内にある各分類子の価値$\dot{S}_j$を式(13), 式(14), 式(15)のように更新する. これらの式は $\dot{\mathrm{Q}}$–Learning における行動価値更新式である式(4)および(3)を, ルールの分類子表現に適用するために拡張したものである. そのため, $\dot{\mathrm{Q}}$–Learning と異なり$k = 1,2 \ldots, Ne$となる. 式(15)は現時点で最も選択される可能性の高い行動を表し, 式(14)ではその行動を有する分類子の強度の合計値を表している.

$$\dot{S}_j \leftarrow (1-\alpha)\dot{S}_j + \alpha\left(\left(r + \gamma \dot{S^{(t)}}_{max}\right)\Big/ |A_{t-k}|\right)\dot{\beta}^{k+1} \tag{13}$$

$$\dot{S^{(t)}}_{max} = \sum\nolimits_{cl_i \in [M_{t+1}] | a^{(t)}_{max}} \dot{S}_i \tag{14}$$

$$a^{(t)}_{max} = \underset{a}{argmax} \sum\nolimits_{cl_i \in [M_{t+1}] | a} Re\left[\dot{S}_i \bar{I}_t\right] \tag{15}$$

また, ZCS では 1 ステップ前の照合集合内に含まれているが行動集合内に含まれていない各分類子に対して選択率を低下させるため, パラメータ$\boldsymbol{\tau}$によって価値を減衰させていた. しかし, CVCS では内部参照値に応じてルールの選択率が変化するため, 価値を減衰させることでルールの選択率が増加する場合がある. そのため, CVCS では価値の減衰処理を行わない.

5.3.3 発見部

CVCS は, ZCS と同様に GA より分類子を進化させるが, 個体の価値は内部参照値に応じて変化するため, 親個体の選択率を正しく見積もるためには内部参照値を必要とする. 行動系列を考慮した上で環境に対して適切なルールを持つ個体を親として選択するために, 親の選択はその時刻での内部状態を用いて**[*A*]**から行う. CVCS において GA が実行される条件は, **[*A*]**内の各分類子が GA の発動対象となってから経過したステップ数の平均値がパラメータθ_{GA}の値を上回る場合である. 式(16)に示される親個体選択確率を基に 2 つの親個体を選択後, 親個体と同一の条件部と行動部をもつ子個体を生成し, **[*P*]**内に追加する. このとき, 各個体の選択率を変化させないために, 親個体の価値の絶対値を半分に減らし, 子個体の価値にその値を設定する, 価値の絶対値を半分にする. この際, **[*P*]**の個体数が最大数 N を超えた場合には, **[*A*]**内から式(16)の逆数に従う確

率で個体を選択し，個体を削除する．ただし，**[*A*]**内の個体数が 2 つ以下の場合は親個体として選択された強度の高い個体の削除を防ぎ，選択圧を保持するため，**[*P*]**内から価値の絶対値の逆数を適合度として，ルーレット選択による選択確率から分類子を削除する．

$$P_{i_{t-1}}(cl_i) = \frac{\exp\left(Re\left[\dot{S}_i\overline{\dot{I}_{t-1}}\right]/T\right)}{\sum_{cl_j\in[A]}\exp\left(Re\left[\dot{S}_j\overline{\dot{I}_{t-1}}\right]/T\right)} \tag{16}$$

なお，CVCS では誤った行動文脈を形成する個体を生成するため，# による分類子の一般化や突然変異を適用しない．また，**[*A*]**内に分類子が 1 つしか存在しない場合，親個体と同一の条件部と行動部をもつ子個体を 1 体生成する．

5.4 CVCS のアルゴリズム

前節で述べたメカニズムを備えた CVCS のアルゴリズムを図 17 に示す．4.4 節で述べた ZCS のアルゴリズムとの差異は，分類子の価値が複素数になった点と，GA の発生条件が変化した点である．

```
while (! end of iteration)
  while (! end of problem)
    state ← recognizing an input from environment：環境から情報を知覚
    generate  [M]  ：[P]から state が照合する分類子を選択
    action  ← selecting the action from [M]
    generate  [A]  ：[M]から action が照合する分類子を選択
    reward  ← execute action：環境から報酬を獲得
    parameter update  ：[A]の Strength 更新（強化部）İの更新
    if ( θ_GA < average elapsed time )
      run GA on [P]  ：[P] を対象に GA を実行（発見部）
    end if
    end while
end while
```

図 17　CVCS のアルゴリズム

5.5 AP-CVCS

以上より，CVCS において適用される進化計算の枠組は，実質的に淘汰による不適切な分類子の削除のみとなる．また，**[*A*]**内の個体数が 3 つ以上の場合は**[*A*]**に対して淘汰が行われるため，交叉によって生成された分類子と同じ条件部

および行動部を持つ分類子が淘汰されることとなり，分類子集合は実質的に変化しない．そのため，淘汰による不要な分類子の削除は[P]内からの淘汰が発生するような限られた状況のみでしか発生しないこととなる．

加えて，CVCS が用いる行動選択の指標では 3.3.2 節に示したような Q̇–Learning と同じ指標によって，(1) 複素行動価値の絶対値の大きさ (2) 複素行動価値の位相と内部参照値の位相の近さという 2 つの観点から行動を決定する．ところが，誤った行動が学習されてしまった場合を考えると，報酬に近づく行動ではないために (1) の指標からは行動選択がなされづらいものの，ひとたび学習されてしまうと前後の行動順の系列に組み込まれることで位相が変化し，(2) の指標から行動選択がなされやすくなる．そのため，誤った行動を適切な行動であると認識してしまうことから，最適解・準最適解の獲得に失敗する場合がある．

そこで，本章では 5 章で述べた CVCS に対して，Population サイズ調整機構および選択圧保護機構を導入することで不要な分類子の淘汰を促し，より少ない試行回数で最適な解を学習可能な手法である AP-CVCS（Adjustment Population size based CVCS）を提案する．AP-CVCS の概念図を図 18 に示す．

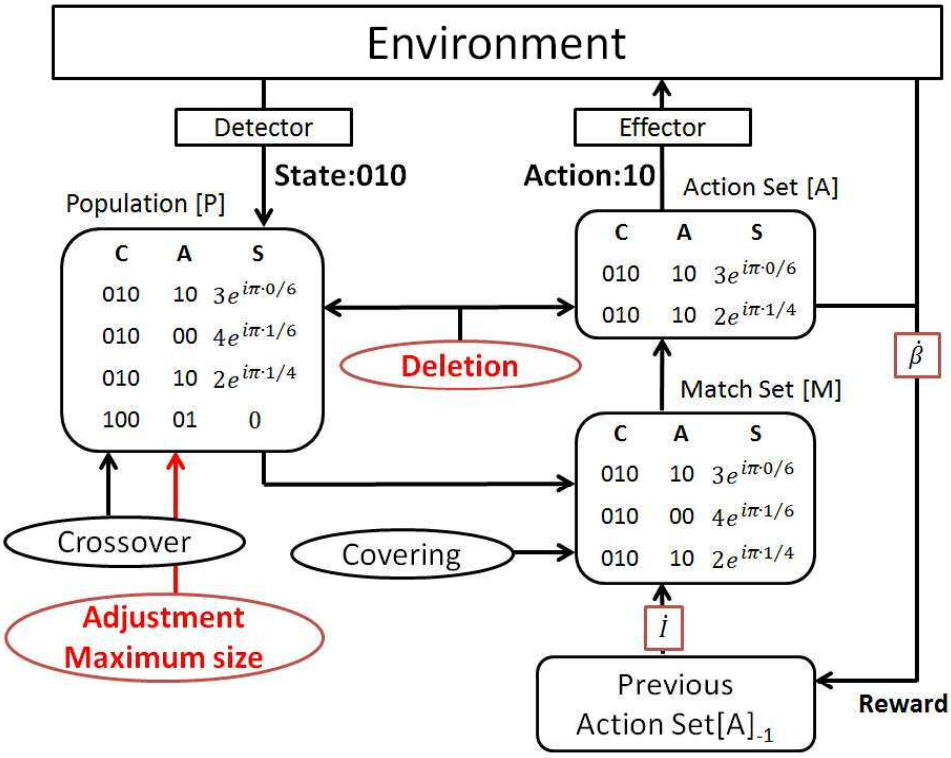

図 18　AP-CVCS の概念図

5.5.1　分類子上限数の調整機構

本節では分類子の状況に関わらず適切な淘汰処理を行うため，CVCS の分類子上限数 N を調整する機構を導入する．具体的には，GA による子個体の生成後，以下に示す 2 つの条件の両方を満たす場合，分類子上限数 N を 1 減らす．

(1) 現在の分類子数が分類子上限数 N を上回る

(2) 現在の分類子数が必要最低限の分類子数以上である

ここで(2)における，必要最低限の分類子数の計算例を図 19 に示す．**[*P*]** 内にある分類子の持つ条件部の種類数と，分類子が取り得る行動部の種類数を掛け合わせた数が，必要最低限の分類子数となる．図の例では条件部の種類数が 3，行動部の種類数が 4 であるため，必要最低限の分類子数は 12 となる．現在の分類子数がこの数を下回る場合，いずれかの状態において特定の行動部を持つ分類子が存在しない状況が発生してしまうため，図 19 に示す数が必要最低限の分類子数となると考えられる．

上述した条件を満たして分類子上限数 N を 1 減らす場合には，GA によって生成された子のうち片方を削除し，**[*P*]** 内に追加しない．これによって，淘汰時に削除する分類子数は 2 で固定となる．

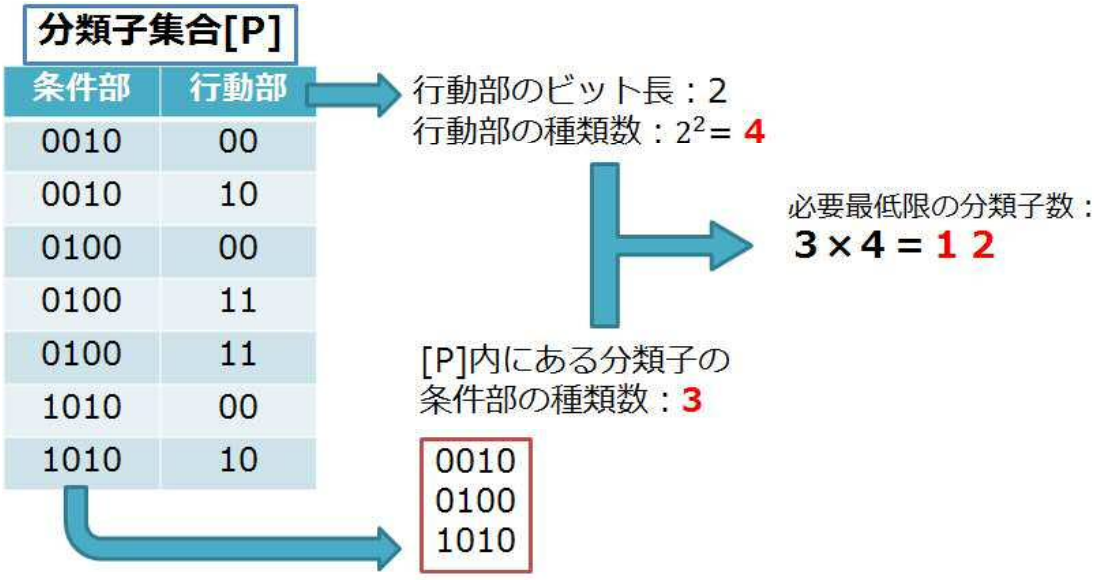

図 19　必要最低限の分類子数

5.5.2　選択圧保護機構

前節で述べた機構の導入によって分類子の状況に関わらず淘汰を行うことができるが，淘汰の促進によって適切なルールが学習する前に削除されてしまう

場合がある．そのような状況に陥ることを防ぐために，行動の選択圧を保護する機構を導入する．具体的には，条件部と行動部がそれぞれ一致する（同一のルールを持つ）個体が他に存在しない分類子を淘汰による削除の対象外とする．この機構を導入した際の削除対象の決定例を図 20 に示す．この機構によって各状態における行動の選択率を保護し，適切なルールの削除（時系列の文脈の破壊）を防ぐことができるようになる．

また，従来では[*A*]内の個体数が 2 つ以下の場合に削除対象を[*P*]内から決定していたが，発見部によって個体が 2 体生成された場合，[*P*]内から同一の条件部および行動部を持つ個体が 2 体同時に削除されることで，削除対象となった個体と同一のルールを持つ個体が[*P*]内に存在しなくなる場合がある．そこで，[*A*]内の個体数が 1 つだけの場合に限って削除対象を[*P*]内から決定することで，行動の選択圧を保護する．

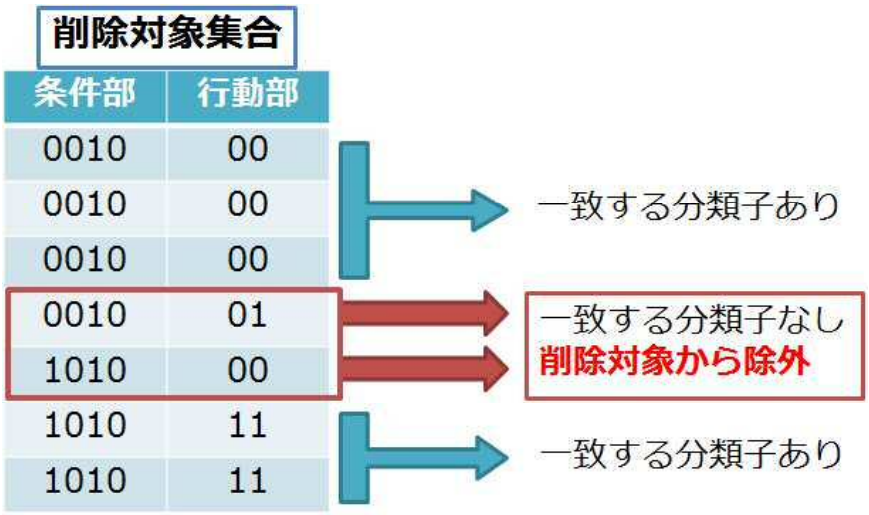

図 20　削除対象の選択例

5.6 AP-CVCS のアルゴリズム

前節で述べたメカニズムを備えた AP-CVCS のアルゴリズムを図 21 に示す．5.4 節で述べた CVCS のアルゴリズムとの差異は，GA を実行した際に Population の最大数を減少させる処理が追加されている点と，選択圧保護のために唯一の行動部および条件部を持つ分類子を淘汰の対象外とする点である．

```
while (! end of problem)
   state ← recognizing an input from environment：環境から情報を知覚
   generate  [M]  ：[P]から state が照合する分類子を選択
   action  ← selecting the action from [M]
   generate  [A]  ：[M]から action が照合する分類子を選択
   reward  ← execute action：環境から報酬を獲得
   parameter update  ：[A]の Strength 更新（強化部）iの更新
   if ( θ_GA < average elapsed time )
      run GA on [P]  ：[P] を対象に GA を実行（発見部）
      if ( |[P]| < N & |[P]| > min. size )
         N ← N – 1
      end if
   end if
end while
```

図 21　AP-CVCS のアルゴリズム

第 6 章　評価問題

POMDPs 環境下における CVCS および AP-CVCS の有効性を評価するため，POMDPs 環境における一般的なベンチマーク問題である Woods 問題を評価問題として用いる．具体的には，1）通常の POMDPs 環境をもつ Woods 問題と 2）難解な POMDPs 環境である，状態空間が大きい Woods 問題と 3) Type 1 と Type 2 の混同が複数混在する Woods 問題を用いる．Woods 問題は多くの学習手法においてテストベッド問題として用いられており[2][21][23], 本論文においてもこの問題を用いることで他の手法との性能比較を容易にする．また，本章では恒常的に不完全知覚問題が発生する基本的な POMDPs 環境の評価問題について紹介し，不完全知覚問題に加えて環境からの外乱などで知覚入力が確率的に変化するような環境については第 9 章にて対応する．

6.1 Woods 問題

Woods 問題は迷路問題の一種であり，学習エージェントは，図 22 のような格子状のフィールド中で食料（Food : "F"）に到達することを目的としている．フィールドは，エージェントが各セル間を遷移し食料（報酬）の獲得を目的とする問題である．フィールドは障害物 (Obstacle : "T"), 通路 (Empty position : " "), Food で構成される．エージェントは現在位置から 8 近傍を状態として知覚可能であり，その 8 近傍に移動可能である．ただし，障害物へ移動した場合は，移動できず現在位置に留まる．提案手法および ZCSM の学習エージェントは，自身の上部から時計回りに，環境中の通路を 00，障害物を 01，食糧を 10 と知覚する．そのため，分類子の条件部は 2*8=16bit で表現されることとなる．エージェントが Food に到達した場合にのみ報酬値rが与えられる．一般的な Woods 問題ではエージェントの初期位置がランダムに決定される．しかし，本研究では，Food 到達までに必ず不完全知覚状態に遭遇させるために，スタート地点 S を初期位置とする．

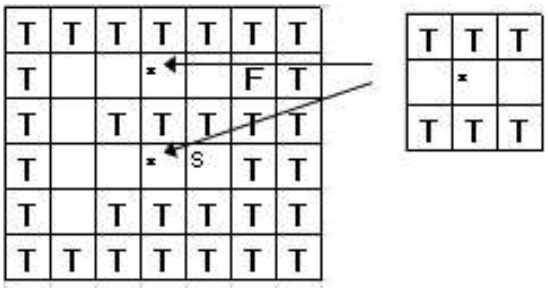

図 22　Woods 問題の例

6.1.1　標準的な POMDPs 環境

図 23，図 24 に，一般的に用いられる POMDPs 環境であり，Type1 に分類される Woods 問題（Type1），Type2 に分類される Woods 問題（Type2）の各フィールドを示す. 図 23 は Littman57[12], 図 24 は MazeF4[17]と呼ばれる問題である．ここで*,**,***は不完全知覚が発生する地点を示している．図 23 のフィールドでは不完全知覚となる状態を何度も経験することで学習機会が多くなるため，他の地点より不完全知覚となる地点の価値が高くなり，適切な方策の獲得が困難な環境である．図 24 のフィールドは，同一状態として知覚される状態において，合理的ルールと非合理的ルールが同一視される問題である．具体的には，初期位置の左のマスでは左に移動する行動，その 2 つ上のマスでは右に移動する行動を獲得しなければ Food に到達することができないという問題がある．なお，本論文では初期位置をランダムに選ぶ必要のある Type 3 の不完全知覚を持つ POMDPs 環境を実験の対象外とした.

T	T	T	T	T	T	T	T	T	T	T	T	T
T	S		*	**	*	**	*					T
T	T	T	***	T	***	T	***	T	F	T	T	T
T	T	T	T	T	T	T	T	T	T	T	T	T

図 23　Type 1 のフィールド

T	T	T	T	T	T	T
T			*		F	T
T		T	T	T	T	T
T			*	S	T	T
T		T	T	T	T	T
T	T	T	T	T	T	T

図 24　Type 2 のフィールド

6.1.2　広大な POMDPs 環境

6.1.1 節で扱う環境よりも，大きな状態行動空間を扱う問題として，図 25，図 26 に示す環境（Type1-Large と Type2-Large）を用いる．この 2 つの問題は，Woods14[2]と呼ばれる問題に対して，POMDPs となるように通路を最右上に追加した問題である．図 25 の環境では Food の 2 つ上の地点の価値の影響を受けて，スタートの左下の地点の行動価値が大きく見積もられることで，誤った方

策が獲得される場合がある．図 26 の環境では，2 回ある不完全知覚のうちスタートに近い地点では下方向に，Food に近い地点では上方向に移動する方策を獲得する必要がある．

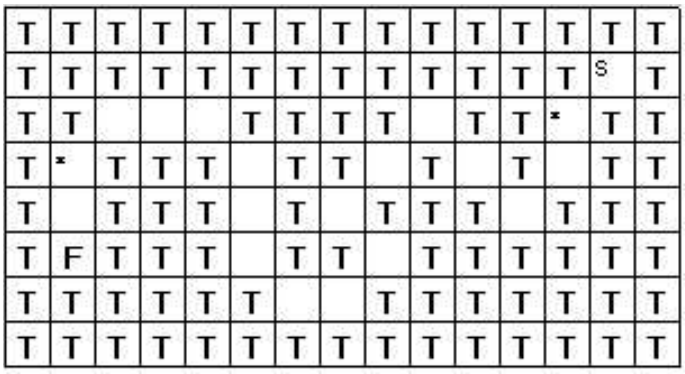

図 25　Type1-Large のフィールド

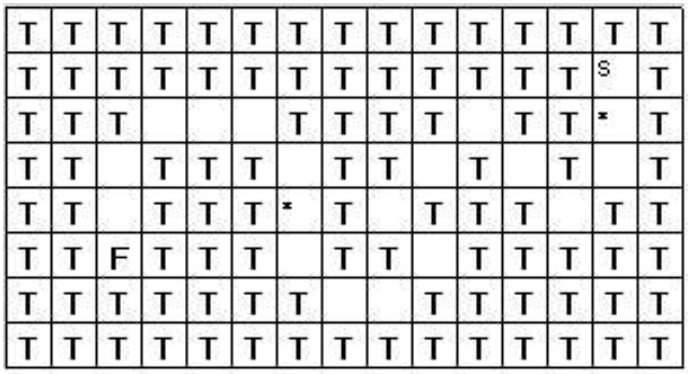

図 26　Type2-Large のフィールド

6.1.3　Type 1 と Type 2 の混同がある POMDPs 環境

Type 1 と Type 2 の混同が複数混在する POMDPs 環境として, 図 27, 図 28, 図 29 に示す環境 (Type2-2, Type1-2, Type2') を用いる． Type2-2 では, Type2 に分類される不完全知覚が 2 回発生する点が特徴である．また, Type1-2 では, *の地点では Type1 の不完全知覚が発生し，**の地点では Type2 の不完全知覚が発生する．最後に，Type2'は図 24 のフィールドと同じ構造をもつが，初期位置が不完全知覚状態となる点で異なる． この問題の難しさとしては，初期位置では時系列による文脈を利用できないため, 最短経路で Food に到達する方策を獲得することが困難となる．

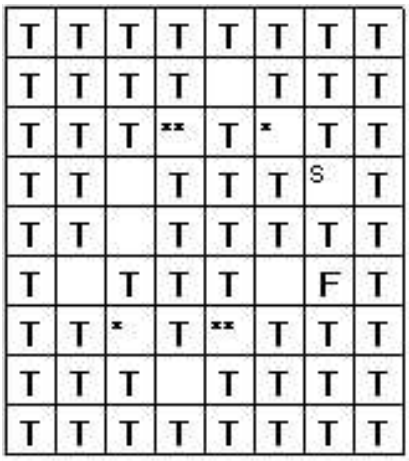

図 27　Type2-2 のフィールド

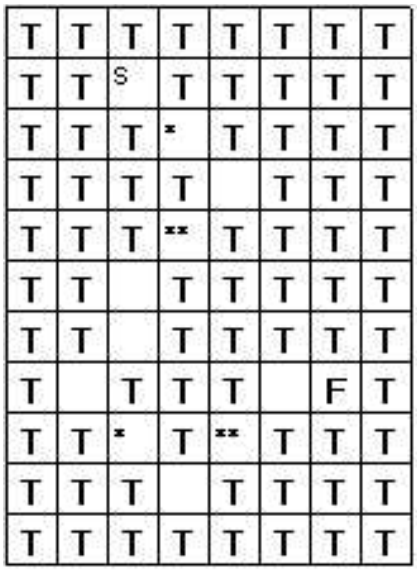

図 28　Type1-2 のフィールド

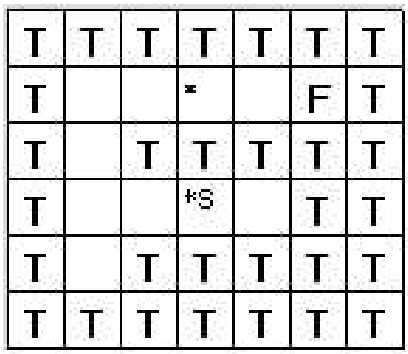

図 29　Type2’ のフィールド

第 7 章　実験 1　CVCS の性能評価

提案手法の有効性を評価するために，第 6 章で説明した 1）通常の POMDPs 環境，2）広大な POMDPs 環境ならびに 3) Type 1 と Type 2 の混同が複数混在する POMDPs 環境における各 Woods 問題に提案手法を適用する．また，後述する評価基準を用いて，従来手法である $\dot{Q}$-Learning と ZCSM と学習性能を比較する．

7.1 評価基準とパラメータ設定

評価基準としては，50 試行での(1)エージェントが Food に到達するまでのステップ数の平均値の推移，および(2)エージェントが Food に到達するまでのステップ数の試行終了時における分布を比較する．学習エージェントは Food に到達することで環境から正の報酬を受け取るため，(1)の評価基準では，Food に到達するまでのステップ数が低い値であるほど，フィールドにおいて適切な方策を獲得できていることを意味する．また，Food に到達するまでのステップ数の収束がより少ない学習回数で達成できるほど，学習に必要な計算コストが少ない手法であることを意味する．同様に，(2)の評価基準では，Food に到達するまでのステップ数が最短ステップに近い試行が多いほど，より確実に適切な方策を獲得できる手法であることを意味する．

ここで，エージェントが 500 ステップ経過後も Food に到達していない場合は，学習を終了し，ステップ数を 500 と設定する．1 試行あたり，10000 回学習を行い，(1)の評価基準は学習回数 200 回ごとの移動平均で示す．また，(2)の評価基準は累積頻度の形で示す．

CVCS，ZCSM および $\dot{Q}$-learning で用いる各パラメータ設定を表 1 に示す．各手法における α, γ, r および $\dot{Q}$-learning の各パラメータについては文献[4]，ZCSM の各パラメータについては文献[2]と同様の設定としている．

また，CVCS および $\dot{Q}$-learning では，問題ごとに，異なる基本位相 $\dot{\beta}$ を設定する（表 2）． CVCS および $\dot{Q}$-learning では行動価値の初期値 S_0 を設定することで，全ての方策が同じ位相を持つことになり，誤った文脈が表現されることから，初期値 S_0 を 0 に設定している．加えて，適切な行動を選択するために CVCS と $\dot{Q}$-learning で異なる温度定数 T を設定している．また，問題ごとに異なる基本位相を用いる理由としては， CVCS および $\dot{Q}$-learning では，1 ステッ

プごとに基本位相だけ回転させた行動価値を学習するため，不完全知覚において Type1 のように同一の行動を取るべき地点では行動価値の位相が同一になるように基本位相を設定する必要があるためである．同様に，Type2 のように異なる行動を取るべき地点では行動価値の位相が真逆となるように，基本位相を設定する．

表 2 に示すような，問題に対して適切な$\dot{\beta}$の位相は，その問題下で不完全知覚となる 2 地点の間を移動するために必要な最短行動数から算出できる．ここで最適な方策を学習した際の，標準的な Type1 および Type2 の POMDPs 環境における価値の位相の伝搬の様子を図 30，図 31 に示す．行動価値関数は報酬獲得時から 1 ステップごとに$\dot{\beta}$ずつ回転して伝搬されることとなる．第 2 章で述べたように，Type 1 では不完全知覚状態において同じ行動を取る必要があるため，全ての不完全知覚状態で同じ位相が伝搬されるよう基本位相を設定することで適切な学習を実現できる．そのため，適切な基本位相は 360° を Type1 の不完全知覚となる 2 地点の間を移動するために必要な最短行動数で割った位相となる．図 30 の例では不完全知覚間の最短行動数が 2step であるため，基本位相は$\exp(i\pi/1)$となる．また，Type 2 では不完全知覚において異なる行動を取る必要があるため, 2 つの不完全知覚に対して伝搬される位相は正反対となるよう基本位相を設定することで, 2 つの不完全知覚状態における適切な行動を区別して学習することができる．そのため，適切な基本位相は 180° を Type2 の不完全知覚となる 2 地点の間を移動するために必要な最短行動数で割った位相となる．図 31 の例では不完全知覚間の最短行動数が 4step であるため，基本位相は$\exp(i\pi/4)$となる．

しかしながら，Type1-2 に関しては，Type1 の不完全知覚に該当する 2 地点の間を移動するためには最短で 6 行動が必要となるが，Type2 の不完全知覚に該当する 2 地点の間を移動するためにも同じく最短で 6 行動が必要となる．Type1 を基準に適切な基本位相を計算すると$\exp(i\pi/3)$となるが，Type2 を基準に適切な基本位相を計算すると$\exp(i\pi/6)$となり，これらは全く正反対の位相となる．そのため，このどちらかに基本位相を設定すると，いずれかの不完全知覚において適切な位相と正反対の位相が設定されることとなり，複素行動価値による行動順が表現できなくなる. そこで本論文では, Type1-2 に対して Type1, Type2 両方の不完全知覚において, 2 地点で伝搬される行動価値の位相差が 90° となるように基本位相を設定している．

表 1　各手法におけるパラメータ

Parameter	CVCS	ZCSM	$\dot{Q}$-learning
α	0.1	0.1	0.1
γ	0.9	0.9	0.9
r	100	100	100
S_0	0	20	0
θ_{GA}	20	-	-
b	-	1	-
φ	-	0.5	-
ρ	-	0.25	-
μ	-	0.5	-
χ	-	0.002	-
$P_{\#}$	-	0.33	-
N	800	800	-
τ	-	0.9	-
Ne	2	-	2
T	0.1	-	1

表 2　各問題における基本位相

Probrem	$\dot{\beta}$
Type1-Small	$\exp(i\pi/1)$
Type2-Small	$\exp(i\pi/4)$
Type1-Large	$\exp(i\pi/4)$
Type2-Large	$\exp(i\pi/4)$
Type2-2	$\exp(i\pi/6)$
Type1-2	$\exp(i\pi/4)$
Type2'	$\exp(i\pi/4)$

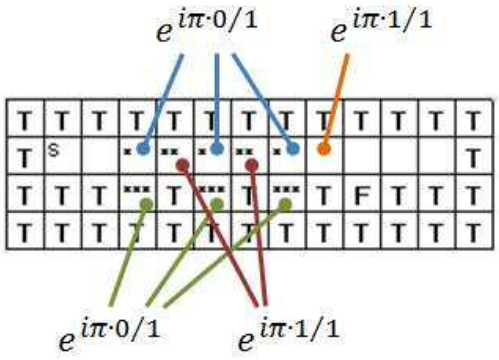

図 30　Type 1 における位相の伝搬の様子

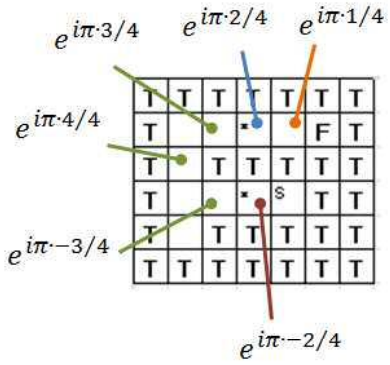

図 31　Type 1 における位相の伝搬の様子

7.2 標準的な POMDPs 環境

図 32，図 33 に Type1 と Type2 における各手法のステップ数を示す．各図において横軸は学習回数，縦軸は学習終了時（Food 到達時）のステップ数である．赤，青，緑のマーカー付の線はそれぞれ CVCS，$\dot{Q}$-Learning，ZCSM の結果を示し，青の太い実線は Food までの最短ステップを示す．CVCS は Type1 と Type2 ともに $\dot{Q}$ -Learning と同程度の学習回数で，ステップ数が収束している．CVCS と ZCSM を比較すると，Type1 では CVCS の収束結果がわずかに劣るが，Type2 においては ZCSM が大きく劣る結果となった．

また，図 34，図 35 に Type1 と Type2 における各手法の最終的なステップ数の分布を示す．赤，青，緑のマーカー付の線はそれぞれ CVCS，$\dot{Q}$-Learning，ZCSM の結果を示し，各図において横軸は最終学習時のステップ数，縦軸は累積頻度である．各図において横軸の最左の数値は最短ステップを示している．これに関しても，Type1 では ZCSM が全体的に良好な傾向となり，Type2 では全体的に悪い傾向にあるものの，CVCS および $\dot{Q}$ –Learning の結果に大きな差は確認できなかった．

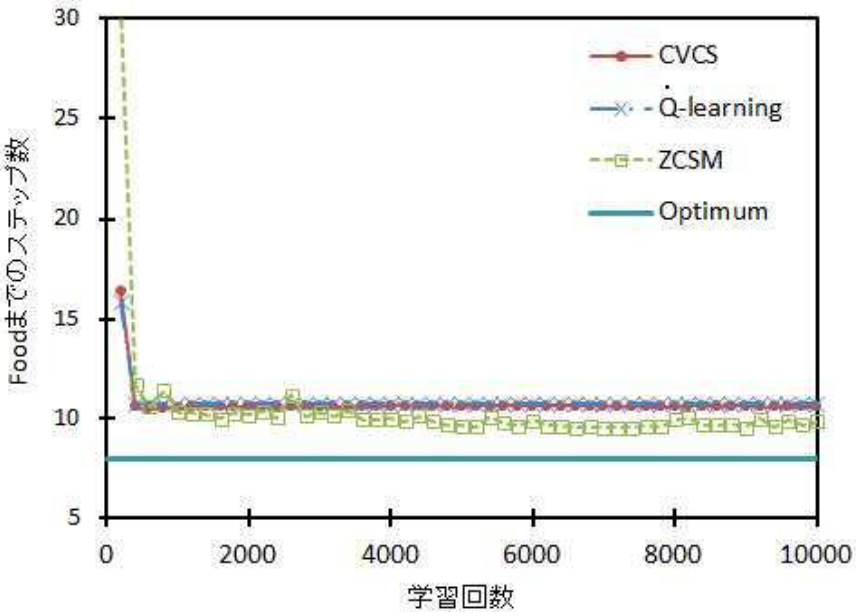

図 32　Type1 の学習結果

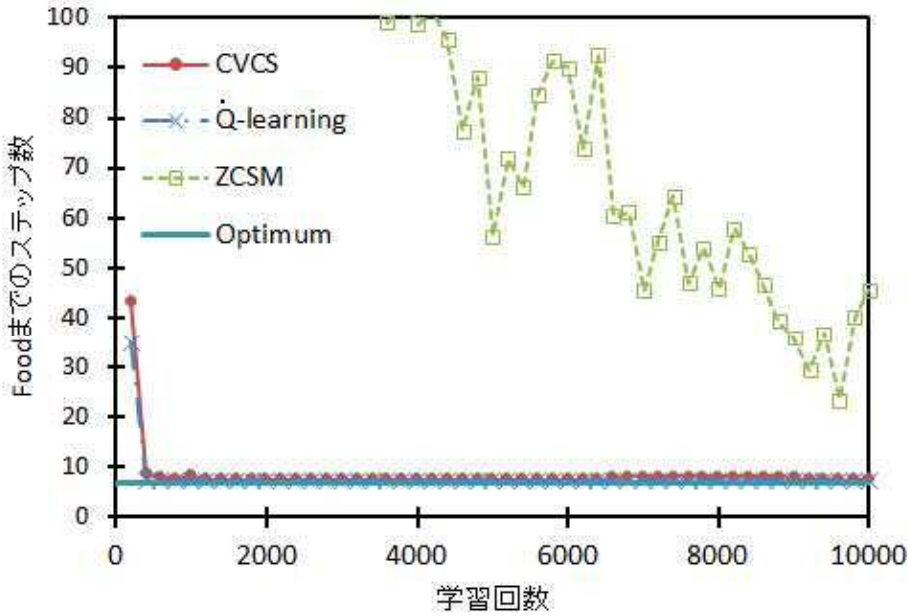

図 33　　Type2 の学習結果

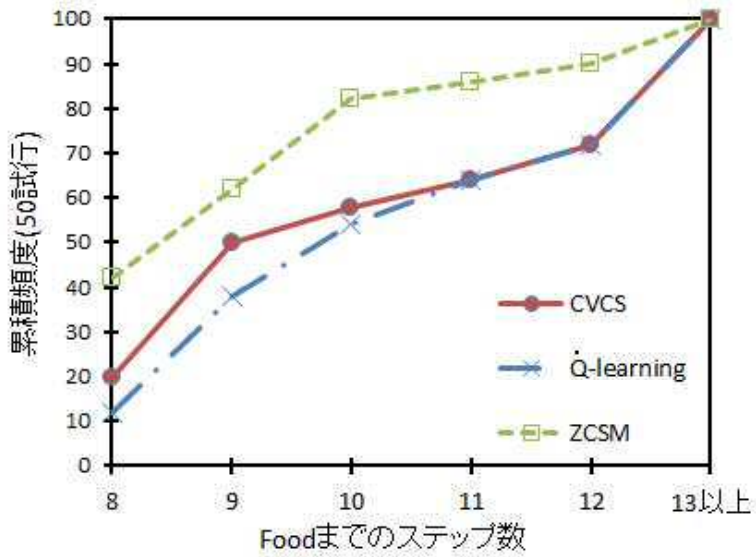

図 34　Type1 の学習結果の分布

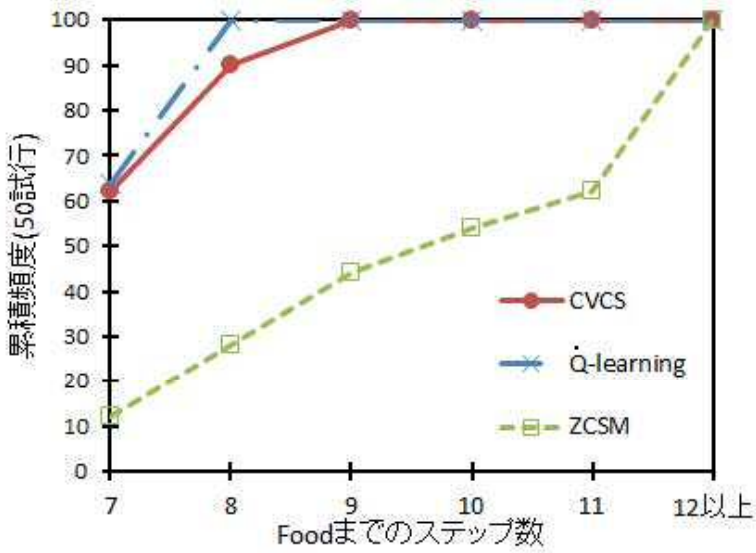

図 35　Type2 の学習結果の分布

7.3 広大な POMDPs 環境

図 36，図 37 に Type1-Large と Type2-Large における各手法のステップ数を示す．各図において横軸は学習回数，縦軸は学習終了時（Food 到達時）のステップ数である．赤，青，緑のマーカー付の線はそれぞれ CVCS，Q̇-Learning，ZCSM の結果を示し，青の太い実線は Food までの最短ステップを示す．両フィールドにおいて，ZCSM は適切な方策を獲得できず，学習に失敗していることがわかる．また，CVCS は，Q̇ -learning と比較して収束したステップ数は同程度である

ものの，少ない学習回数でステップ数が収束していることがわかる．

また，図 38，図 39 に Type1-Large と Type2-Large における各手法の最終的なステップ数の分布を示す．各図において横軸は最終学習時のステップ数，縦軸は累積頻度である．赤，青，緑のマーカー付の線はそれぞれ CVCS，$\dot{Q}$-Learning，ZCSM の結果を示し，各図において横軸の最左の数値は最短ステップを示している．この図から，ZCSM と比較して CVCS，$\dot{Q}$ –learning の両手法が同等に良好な結果を示していることが確認できる．

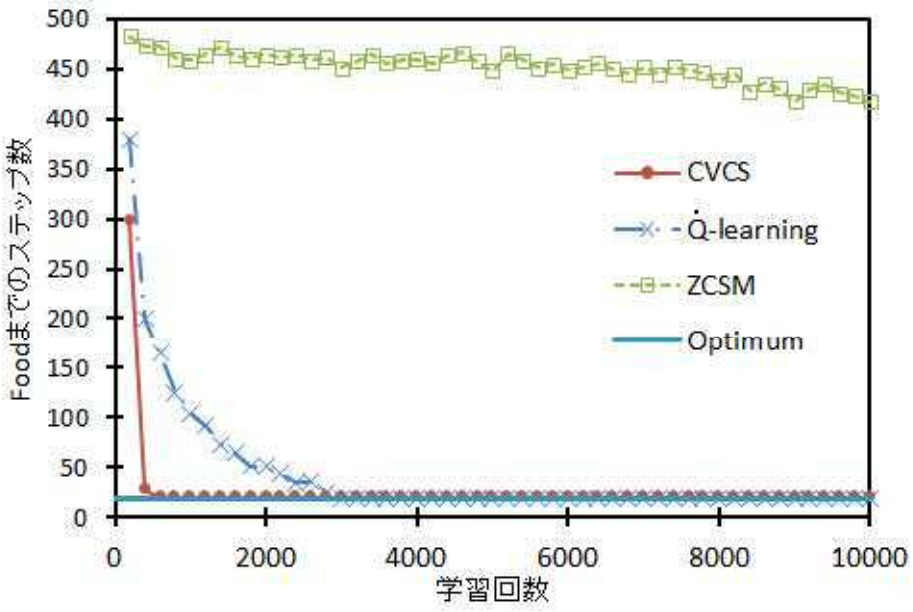

図 36　Type1-Large の学習結果

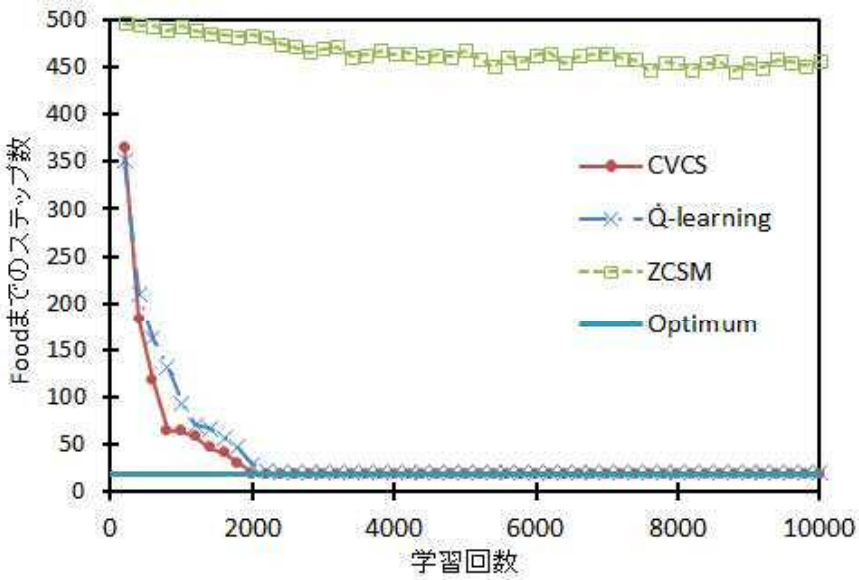

図 37　Type2-Large の学習結果

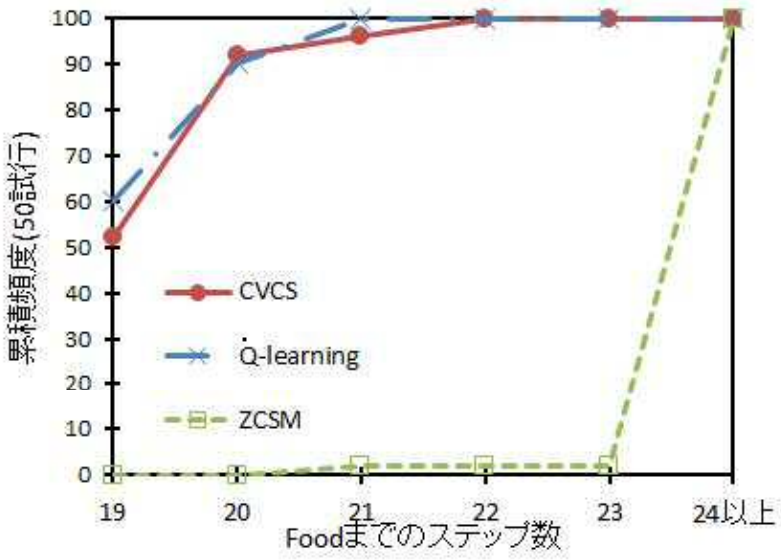

図 38　Type1-Large の学習結果の分布

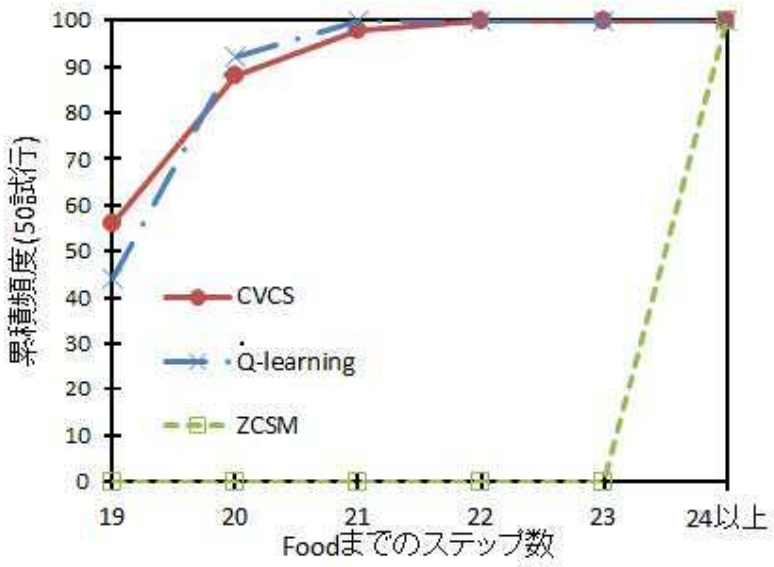

図 39　Type2-Large の学習結果の分布

7.4 Type 1 と Type 2 の混同がある POMDPs 環境

図 40，図 41，図 42 に，Type2-2，Type1-2，Type2'における各手法のステップ数を示す．各図において横軸は学習回数，縦軸は学習終了時（Food 到達時）のステップ数である. 赤, 青, 緑のマーカー付の線はそれぞれ CVCS, Q̇-Learning, ZCSM の結果を示し, 青の太い実線は Food までの最短ステップを示す. 図より, ZCSM は全フィールドにおいて学習不可能であることがわかる. 図 40 および図 41 より，不完全知覚が複数発生するような環境においても，CVCS は少ない学習回数でステップ数が適切に収束していることがわかる．同様に，異なる Type の不完全知覚が混在するような環境においても，CVCS は Q̇ -learning より少ない学習回数でステップ数が収束している．特に Type1-2 では，不完全知覚に対して適切な基本位相が設定できない問題であり，Q̇-learning では学習結果の収束に多くの時間を必要とする困難な問題である．しかし，CVCS では安定してステップ数が収束していることがわかる．一方，Type2’では CVCS と Q̇-learning は同等の学習結果を示している.

また，図 43，図 44，図 45 に Type2-2，Type1-2，Type2'における各手法の最終的なステップ数の分布を示す. 赤, 青, 緑のマーカー付の線はそれぞれ CVCS, Q̇-Learning，ZCSM の結果を示し，各図において横軸は最終学習時のステップ数，縦軸は累積頻度である．各図において横軸の最左の数値は最短ステップを示している. Type2-2 および Type1-2 では, これまでの学習結果と同様に, ZCSM の結果が大きく劣り，CVCS および Q̇ –learning が同等の結果となっている.

しかしながら，Type2’においては Type2 と同等のフィールド構造を有しているにも関わらず，CVCS および ZCSM が最短ステップと同じ解を一定以上の割合で獲得する一方，$\dot{Q}$-learning は全ての試行において 9 ステップで Food に到達する行動方策を獲得している．

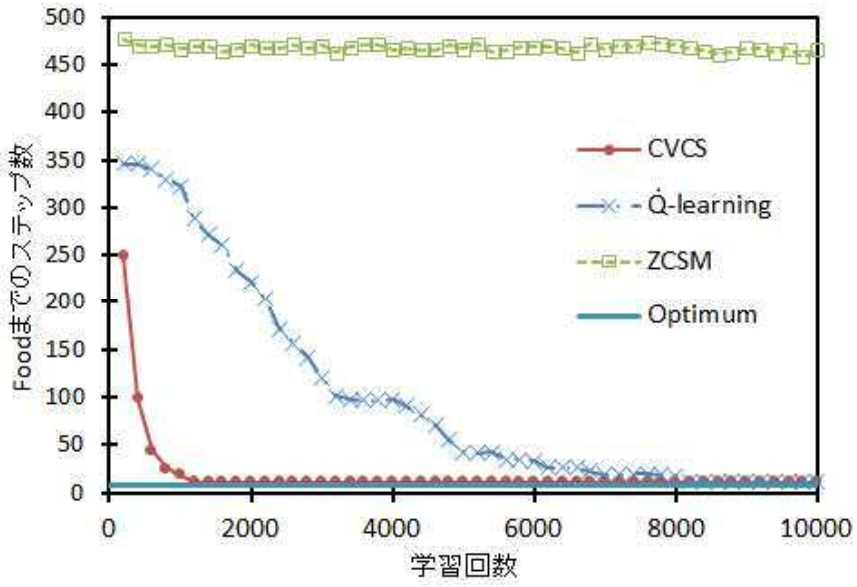

図 40　Type2-2 の学習結果

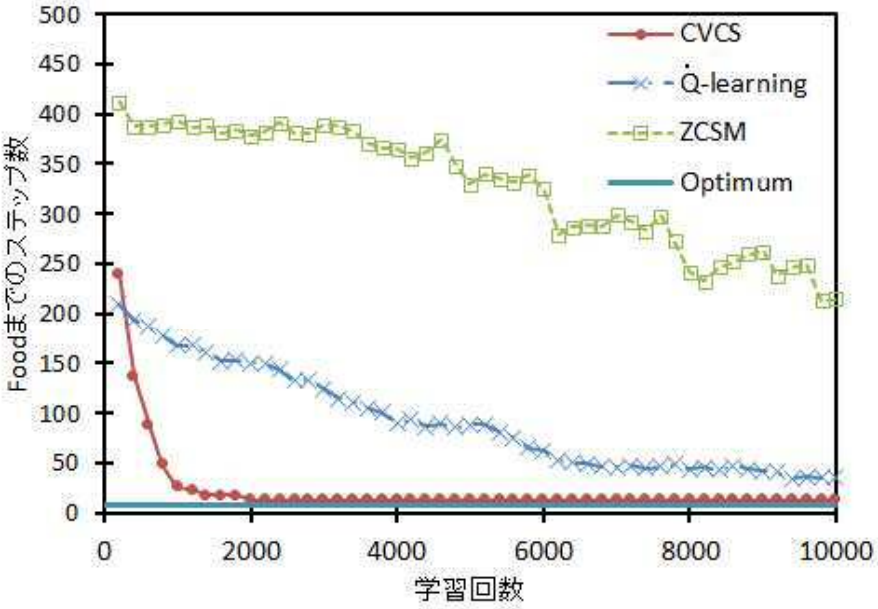

図 41　Type1-2 の学習結果

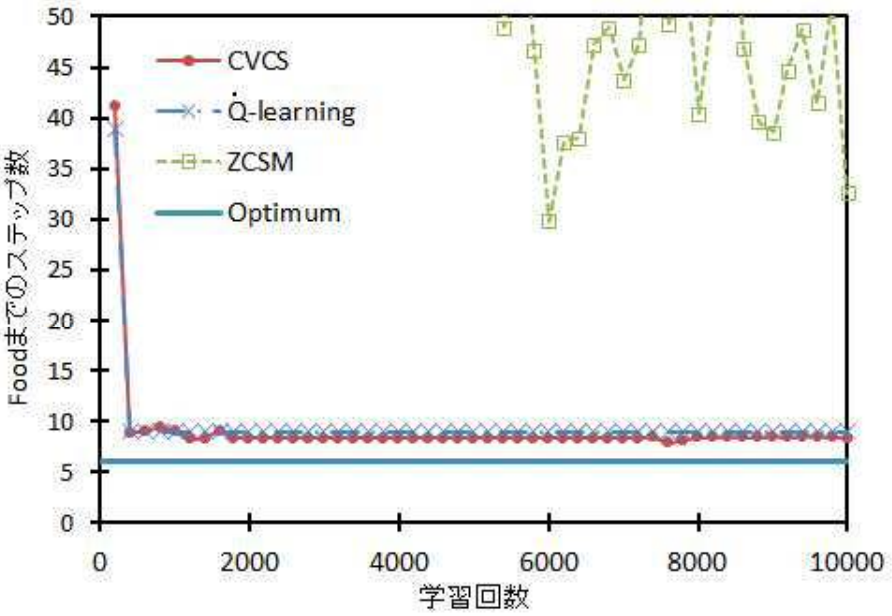

図 42　Type2’の学習結果

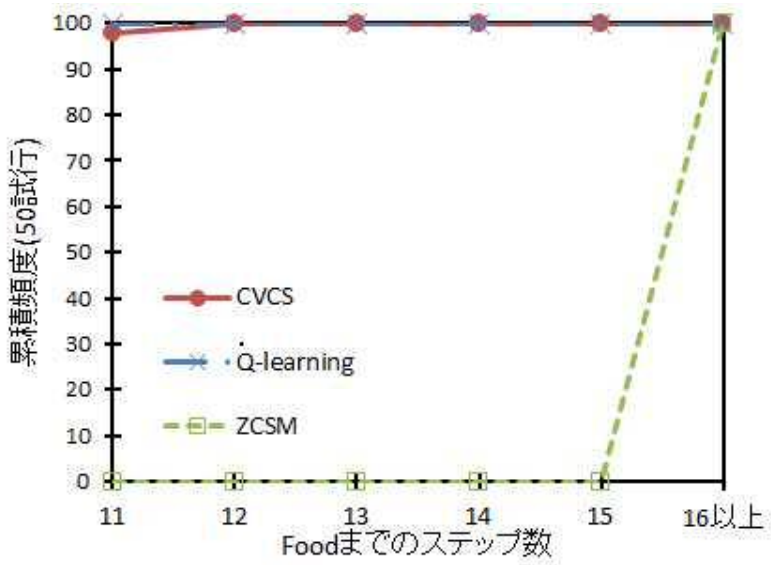

図 43　　Type2-2 の学習結果の分布

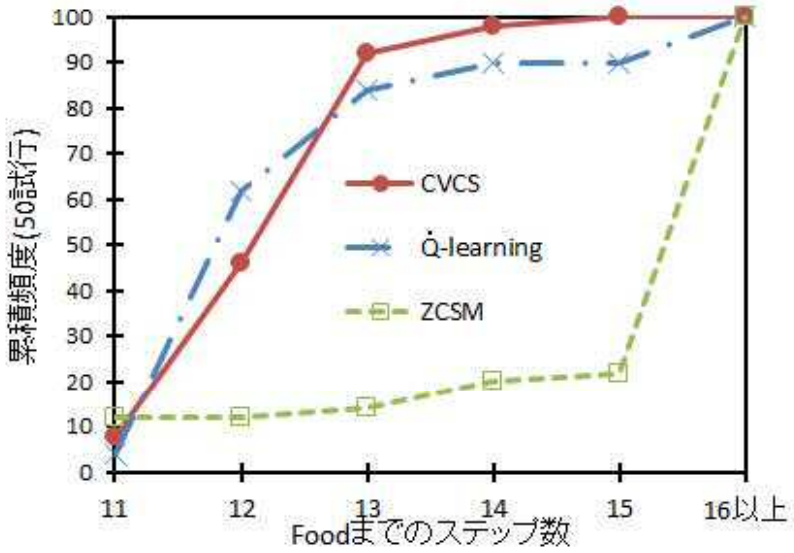

図 44　Type1-2 の学習結果の分布

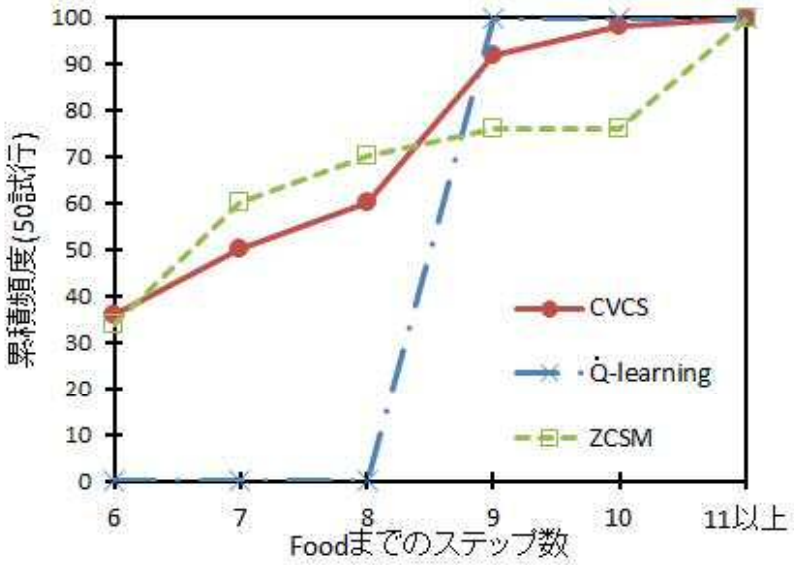

図 45　Type2’の学習結果の分布

7.5 考察

7.5.1　提案手法と $\dot{Q}$ –Learning の比較

提案手法は評価基準(1)学習回数に対する Food までのステップ数の推移において，Type1-2 を除く全ての環境で $\dot{Q}$ -Learning と同程度の収束結果を示している．このような結果となった原因は，提案手法と $\dot{Q}$ -Learning が不完全知覚状態の識別に複素数による行動価値という同じ枠組を用いているためであると考え

られる．提案手法では $\dot{Q}$ -Learning と異なり，分類子によって方策を表現しているが，一般化や突然変異を行わない事から獲得可能な方策としては両手法とも実質的に同等である．

評価基準(2)である最終的な Food までのステップ数の累積頻度からの考察としては，図 44 から確認できるように，Type1-2 では提案手法が全ての試行で 15 ステップ以内に Food に到達する行動方策を獲得できた一方，$\dot{Q}$ -Learning では 50 試行中 5 試行（10%）の試行で，Food に到達するための方策を獲得できなかった．このように Type1-2 の学習結果に差が発生した原因については，Type1-2 が時系列の文脈を構築しづらい特殊な構造をしていることが考えられる．
Type1-2 では Type1 の不完全知覚と Type2 の不完全知覚の両方が存在しており，7.1 節で述べたとおり，各不完全知覚となる状態について行動価値の差が 90° となるように基本位相を設定している．そのため，Type1 の不完全知覚状態では 90° 異なる 2 種類の位相を持つ価値が伝搬されることとなり，それぞれの地点で適切な位相を持つことができなくなることから，誤った行動を選択する原因となる．また，Type2 の不完全知覚状態では各地点で選択すべき行動価値の差が 90° しかないため，他の行動価値との位相差が小さくなることで間違った行動を選択する状況が発生する．これらの要因によって，Food に到達するための方策を獲得できない試行が存在することとなる．

実際に得られた実験例からの具体例を図 46 およびによって説明する．は Type1-2 において不完全知覚によって時系列の文脈が破壊した際の各状態における行動価値，は問題環境と状態との対応図である．図から，Type1 の不完全知覚となる状態 A の価値には基本位相$\dot{\beta}$の値によって，状態 B および C の複素行動価値から0.25πだけ回転した値が伝搬される．最適な行動を選択している場合には状態 B で左下に行動する価値と状態 C で右上に行動する価値が伝搬されるが，状態 B と比較して状態 C は目標状態に近いため，行動価値の絶対値も大きくなり，状態 A に伝搬される行動価値も大きくなる．そのため，状態 B の位相から$\dot{\beta}$だけ回転した位相0.429πよりも状態 C の位相から$\dot{\beta}$だけ回転した位相1.0πに近い値として，状態 A から右下に移動する行動価値の位相は0.893πに接近する．これによって，状態 A の行動価値と状態 B から左下に移動する行動価値は基本位相$\dot{\beta}$の値よりも大きく離れることとなり，異なる位相を持つ行動が選択されやすくなる．表の例では t=7685 において式(7)によるボルツマン選択から状態 B において左上に移動する行動が選択されたことから誤った行動価値の伝搬によって時系列の文脈が破壊され，t=7686 では状態 A と状態 B を交互に移動する確率が非常に高い方策となってしまい，学習に失敗している．このように，不完全知覚に対して適切な位相を設定できないことで学習に失敗するケースが発生する．一方，CVCS ではこのようなケースは確認されなかった．これは淘汰の

枠組によって，表の例でいう状態 B において左上に移動する行動のような価値が減衰することで，各行動価値が適切な位相とならない場合でも適切な行動価値とそうでない行動価値の絶対値に差が開き，誤った行動が選択される確率が大きく減少するためである．

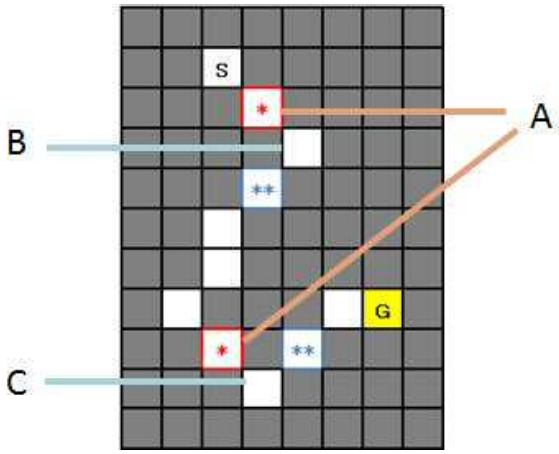

図 46　Type1-2 のフィールドおよび状態の割り当て

表 3　Type1-2 における行動価値の具体例

State	Action	t=7684	t=7685	t=7686
A	右下	$44.6\exp(i\pi\cdot 0.893)$	$39.5\exp(i\pi\cdot 0.897)$	$3.6\exp(i\pi\cdot 0.216)$
B	左上	$10.1\exp(i\pi\cdot -0.911)$	$12.7\exp(i\pi\cdot -0.984)$	$2.8\exp(i\pi\cdot -0.193)$
B	左下	$37.0\exp(i\pi\cdot 0.179)$	$37.0\exp(i\pi\cdot 0.179)$	$37.0\exp(i\pi\cdot 0.179)$
C	右上	$88.0\exp(i\pi\cdot 0.750)$	$88.0\exp(i\pi\cdot 0.750)$	$88.0\exp(i\pi\cdot 0.750)$

7.5.2　提案手法と ZCSM の比較

評価基準(1) 学習回数に対する Food までのステップ数の推移から提案手法と ZCSM の性能を比較すると，ZCSM では Type1 を除いて Type1-Large などのように Food までの最短ステップ数が大きい環境であるほど Food に到達するための方策が獲得できないことが多く，提案手法に大きく劣る結果となった．この理由としては 4.5 節で述べた通り，ZCSM の内部レジスタによって他の手法よりも状態空間が広大になるため，大きい状態空間を扱う場合に探索効率がより低下したと考えられる．表 4 に，Type2-Larg の不完全知覚となる状態における ZCSM が獲得した各行動の強度値合計の具体例を示す．なお，全ての内部行動値において強度値が 0 となる行については割愛した．この不完全知覚では，目標状態に近い地点では上，初期状態に近い地点では下を選択する必要がある．しかし，表から分かる通り，不完全知覚となる状態においてほとんどの行動価値は 0 となり，分類子が存在していない．また，適切な行動の 1 つである上に

移動する行動の強度は内部状態が 0 の場合に他の行動と比較して大きな強度を持つものの，適切な行動の 1 つである下に移動する行動の強度は非常に小さい値であり，内部状態が 1 の場合に選択されることがない．内部状態が 0 の時は高確率で他の行動が選択されることから，内部条件が 1 かつ下に移動する行動を持つ分類子を獲得・学習する必要があるが，内部レジスタによる状態行動空間の増大によってそのような分類子を進化計算によって獲得することは困難となる．本研究の実験ではメモリのビット長を全ての環境で最小の 1 と設定しているため，より大きなメモリが必要となる環境においては更に探索効率が低下する．また，上述した理由から最終的に Food に到達するための方策が獲得できない事が多いため，評価基準(2) 最終的な Food までのステップ数の累積頻度についても Type1 を除く全てのケースで，学習に失敗した試行の割合から提案手法が優れる結果となった．

表 4　ZCSM における行動価値の具体例

内部条件	行動	内部行動：0	内部行動：#	内部行動：1
0	上	0	0	7.130
0	下	0	0	0.742
#	右	0.108	0	0
#	右下	0	2.461	0.085
1	上	0	0	0.002
1	右上	0.138	0	0
1	右下	0	1.894	0
1	左	0.390	0	0

7.5.3　POMDPs の Type による提案手法の特性

Type1, Type2, Type1-Large, Type2-Large の結果から，提案手法は POMDPs の種類や環境の規模に関わらず，従来手法と同等の性能を有した方策を獲得可能であることが分かる．また，Type1-Large や Type2-2 などいくつかの結果において提案手法の学習結果の収束が従来手法よりも早い理由に関しては，学習初期において，価値の低い不適切なルールを淘汰することで，不適切なルールの選択率（価値の更新機会）を減少させているためである．具体的には，Type1-Large において学習中と考えられる 1000 回学習時の行動価値関数（分類子集合内）に存在する適切な状態行動ルール（その状態において Food に到達するために必要な行動回数が少なくなる地点に移動可能な行動）の価値（強度値）が占める割合を調査したところ，図 47 に示すように，$\dot{Q}$ –Learning が 70.2%であったのに対し，提案手法では 77.0%という結果となった．この淘汰の働きによって相対的に選択すべきルールの選択率が増加することでより高速な学習が可能となり，学習の収束が促進される．

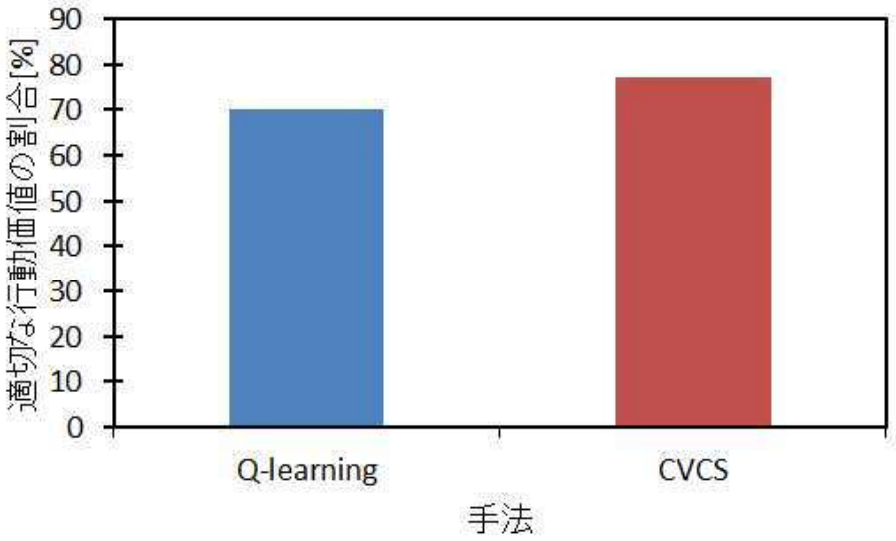

図 47　適切な行動価値の占める割合

7.5.4　複雑な POMDPs 環境における提案手法の特性

Type2'では，評価基準(1)の観点では図 42 からわかるように， CVCS と $\dot{Q}$ -learning では同程度の学習回数でステップ数が収束している．しかしながら，図 45 で示したように，評価基準(2)の観点である最終的なステップ数の分布としては両手法の間に大きな差が確認できた．このような結果が出た原因としては，式(6)および式(11)に示される，各手法の初期状態における内部参照値$\dot{i}_{-1}$の決定方法の違いが考えられる．

各手法における Type2’で獲得された各ルールの価値の絶対値および選択される初期内部参照値の例を図 48 に示す．$\dot{Q}$-learning における初期内部参照値は，初期位置の状態において最も絶対値の大きい複素行動価値であり，不完全知覚となる初期状態では Food の報酬がより強く伝搬される右移動の行動の価値が最も強くなる．そのため，本来選択するべき左移動の行動価値の位相は正反対となることで，Food から遠ざかる右移動を選択し最適な方策を獲得することができない．

一方，CVCS では行動単位でなく，強度の絶対値が最も大きなルールの強度と同じ位相が初期内部参照値となる．また，式(13)に示すように，提案手法において報酬値が伝搬される際，報酬値が更新対象となる**[*A*]**の要素数で割られてから各分類子に伝搬するため，本来の報酬値は**[*A*]**内の各分類子に対して均一に分配されることとなる．そこで，行動部が右移動であるルールを多数保持し，一方で行動部が左移動であるルールを少数保持することによって，複数のルールに報酬が分配される右移動ルールに比べて左移動ルールの強度が大きな値となる．

これによって初期状態において左移動のルールの強度が内部参照値として選択され，最短経路で Food に到達する方策を獲得することができる．ルールの強度の合計では $\dot{Q}$-learning と同様に右移動のルールの強度が最も高くなるが，多く選択される行動ほど親個体として選択されやすくなり，GA によって複数の小さい強度を持つ分類子になる傾向がある．また，報酬に近いルールは学習が早いため，他のルールに比べて親個体として選択される割合は増加する．このような理由によって右移動ルールの分類子数が増加することで最適な方策を獲得することが可能となる．しかしながら，初期状態において確実に正しい内部参照値が選択される保障はないために学習が不安定となり，平均的な学習結果としては図 42 のように CVCS と $\dot{Q}$-learning で同等の結果となっている．

最後に，第 7 章における実験結果のまとめとして，各環境における各手法の性能を示した表を表 5 に示す．表のマークは◎を最良として，○，△，×の順に良好な結果であることを示している．評価基準(2)では CVCS と $\dot{Q}$-learning の間に大きな差が確認できなかったため，ここでは評価基準(1)において，学習終了時の平均ステップ数および学習の速度（学習結果の収束までにかかったステップ数）から評価を行っている．表から分かる通り，提案手法は従来手法と比較してさまざまな環境で良好な結果を示していることが分かる．

図 48　初期内部参照値の選択例

表 5　第 7 章における実験結果のまとめ

Environment	CVCS	ZCSM	$\dot{Q}$-learning
Type1	◎	◎	◎
Type2	◎	×	◎
Type1-Large	◎	×	○
Type2-Large	◎	×	◎
Type2-2	◎	×	○
Type1-2	◎	×	△
Type2'	◎ 最適解を獲得可能	×	◎ 安定した学習

第 8 章　実験 2 AP-CVCS の性能評価

5.5 節で述べた改良を加えた提案手法の有効性を評価するために，第 6 章で説明した 1）広大な POMDPs 環境ならびに 2) Type 1 と Type 2 の混同が複数混在する POMDPs 環境として，Type1-Large，Type2-Large，Type1-2，Type2'の各問題に改良後の提案手法を適用する． Type 1 および Type 2 に各手法を適用していない理由としては，これら問題環境の規模が他の問題と比べて小さいために第 7 章の実験結果において手法間の大きな差異が確認できず，本章の実験においても特徴的な結果が確認できないためである．ここでは，第 7 章と同様に 7.1 節で述べた評価基準を用いて，改良前の提案手法と学習性能を比較する．なお，パラメータ設定は 7.1 節で述べたものと同一である．

8.1 広大な POMDPs 環境

図 49，図 50，図 51，図 52 に Type1-Large と Type2-Large における各手法のステップ数を示す．各図において横軸は学習回数，縦軸は学習終了時（Food 到達時）のステップ数である．緑，赤，青のマーカー付の線はそれぞれ AP-CVCS，CVCS，$\dot{Q}$-Learning の結果を示し，青の太い実線は Food までの最短ステップを示す．全体的な推移を見ると， AP-CVCS と CVCS に大きな差は確認されなかった．しかしながら，50 試行の平均ステップ数としては AP-CVCS が最も優れる結果となった．

また，図 53，図 54 に Type1-Large と Type2-Large における各手法の最終的なステップ数の分布を示す. 緑, 赤, 青のマーカー付の線はそれぞれ AP-CVCS, CVCS，$\dot{Q}$-Learning の結果を示し，各図において横軸は最終学習時のステップ数，縦軸は累積頻度である．各図において横軸の最左の数値は最短ステップを示している．学習結果の拡大図と同様に，この評価基準における結果からも AP-CVCS が他の手法と比べて最も優れている結果が確認できた．特に，Type2-Large では 50 試行中全ての試行において最短ステップで Food に到達できる最適解を獲得できていることが確認できた.

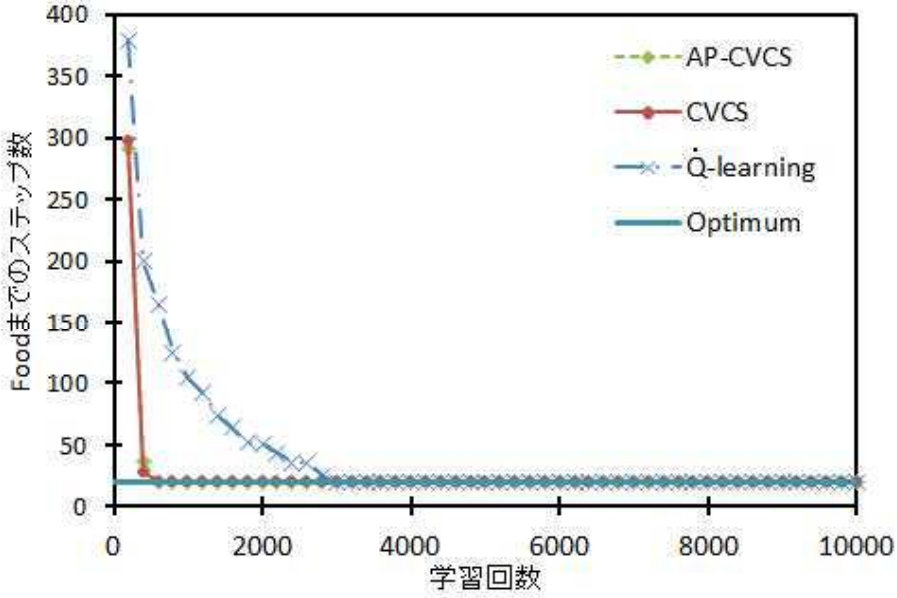

図 49　Type1-Large の学習結果

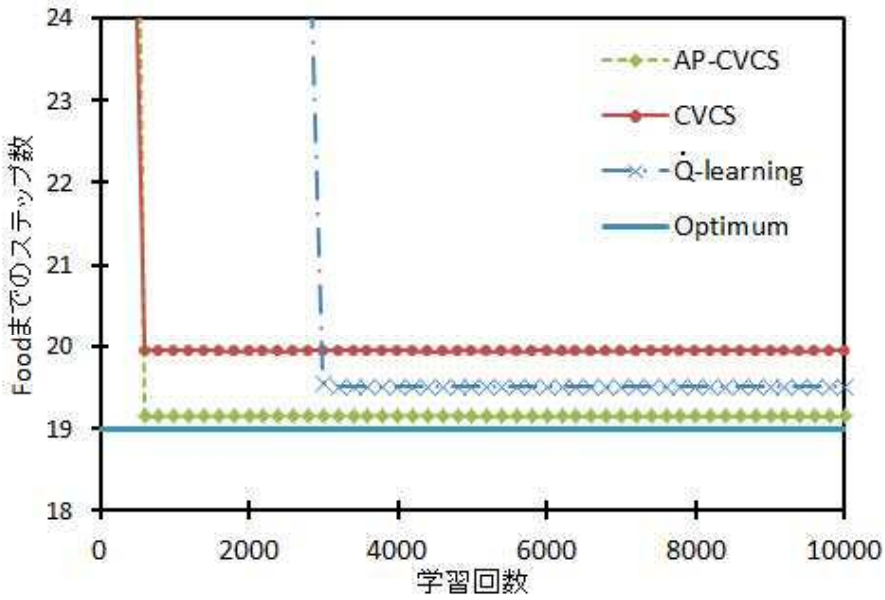

図 50　Type1-Large の学習結果（Optimum 付近の拡大）

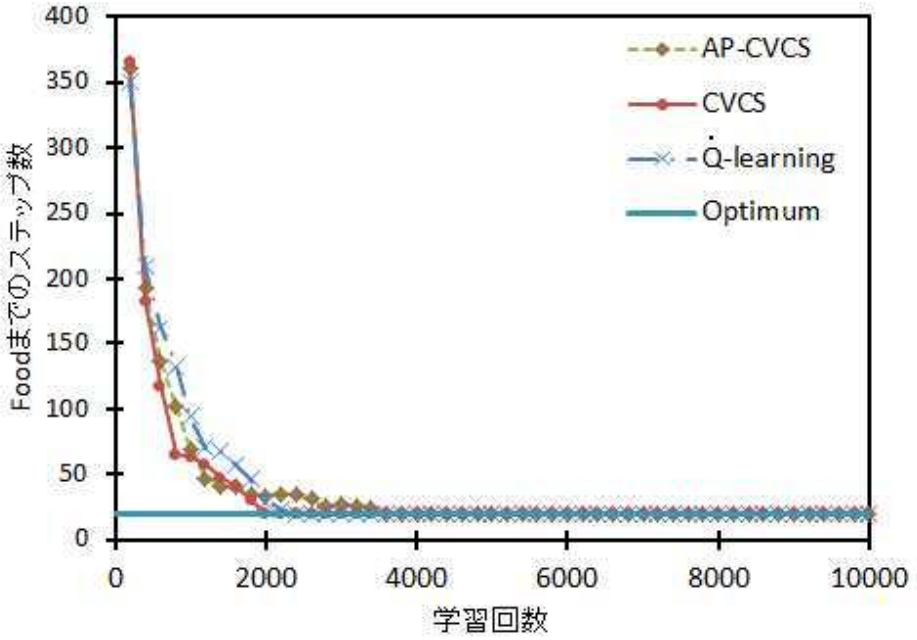

図 51　Type2-Large の学習結果

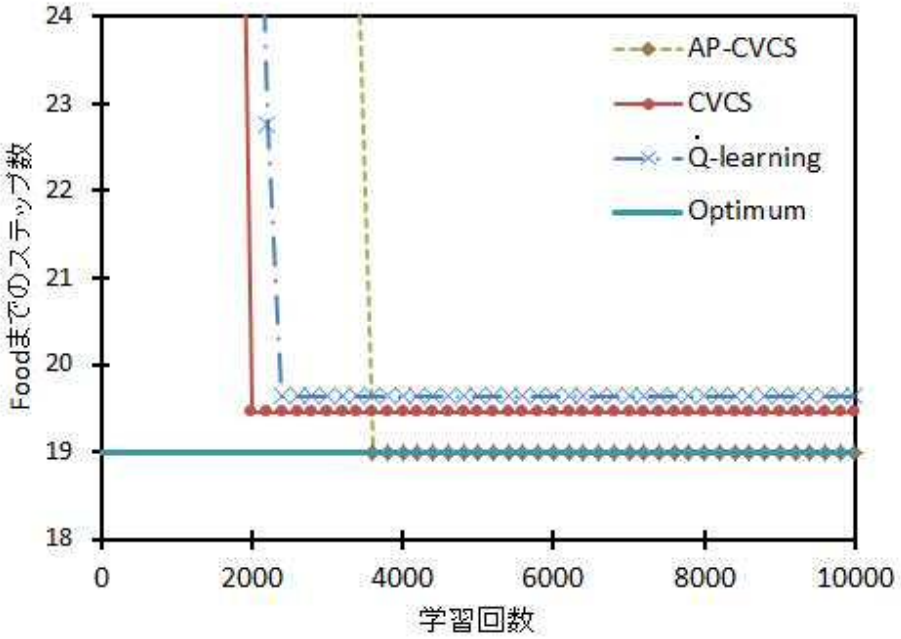

図 52　Type2-Large の学習結果（Optimum 付近の拡大）

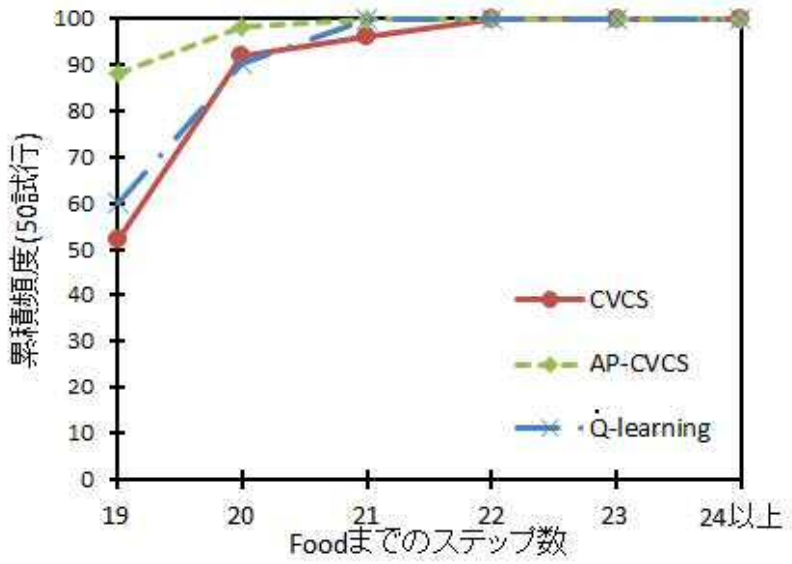

図 53　Type1-Large の学習結果の分布

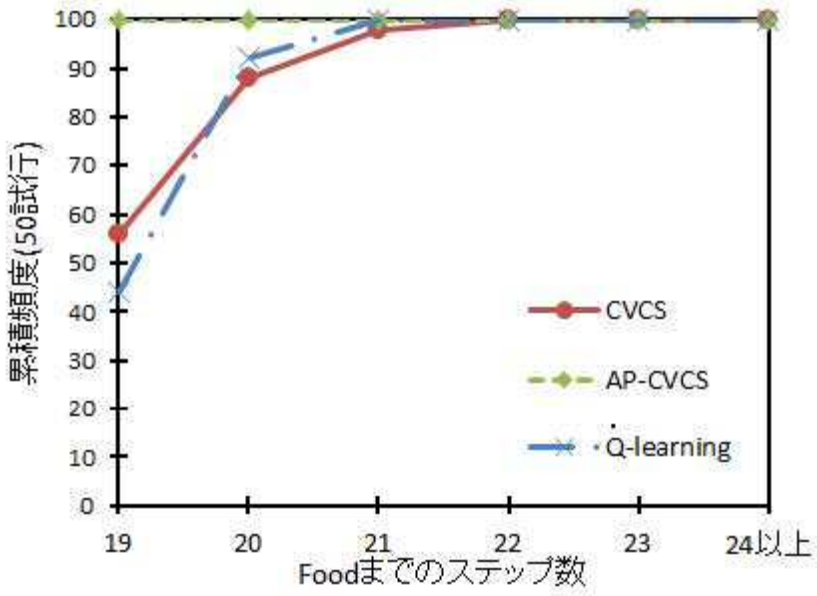

図 54　Type2-Large の学習結果の分布

8.2 Type1 と Type2 が混在する POMDPs 環境

図 55，図 56，図 57，図 58 に Type1-2 と Type2'における各手法のステップ数を示す．各図において横軸は学習回数，縦軸は学習終了時（Food 到達時）のステップ数である. 緑, 赤, 青のマーカー付の線はそれぞれ AP-CVCS, CVCS, Q-Learning の結果を示し，青の太い実線は Food までの最短ステップを示す. Type1-2 では 8.1 節で述べた広大な POMDPs 環境と同様に, AP-CVCS は CVCS

と比較してわずかに収束が遅いものの，平均的なステップ数は最も少ない値となった．一方，Type2’においては，AP-CVCS は試行初期において CVCS と同等の学習傾向を示したものの，次第にステップ数が増加し，最終的には $\dot{Q}$-learning と同程度に収束している．

また，図 59，図 60 に Type1-2 と Type2’における各手法の最終的なステップ数の分布を示す．緑，赤，青のマーカー付の線はそれぞれ AP-CVCS，CVCS，$\dot{Q}$-Learning の結果を示し，各図において横軸は最終学習時のステップ数，縦軸は累積頻度である．各図において横軸の最左の数値は最短ステップを示している．AP-CVCS は Type1-2 においては 他の手法と比べて全体的に良好な結果となっているが，Type2’においては $\dot{Q}$-Learning と同様に，全ての試行において Food に 9 ステップで到達する解を獲得していることが確認された．

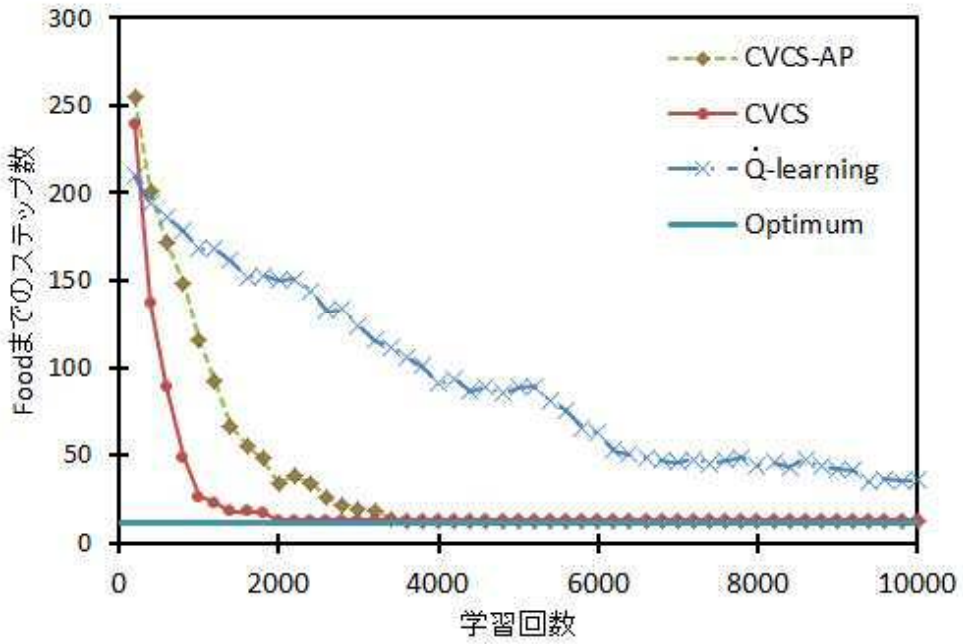

図 55　Type1-2 の学習結果

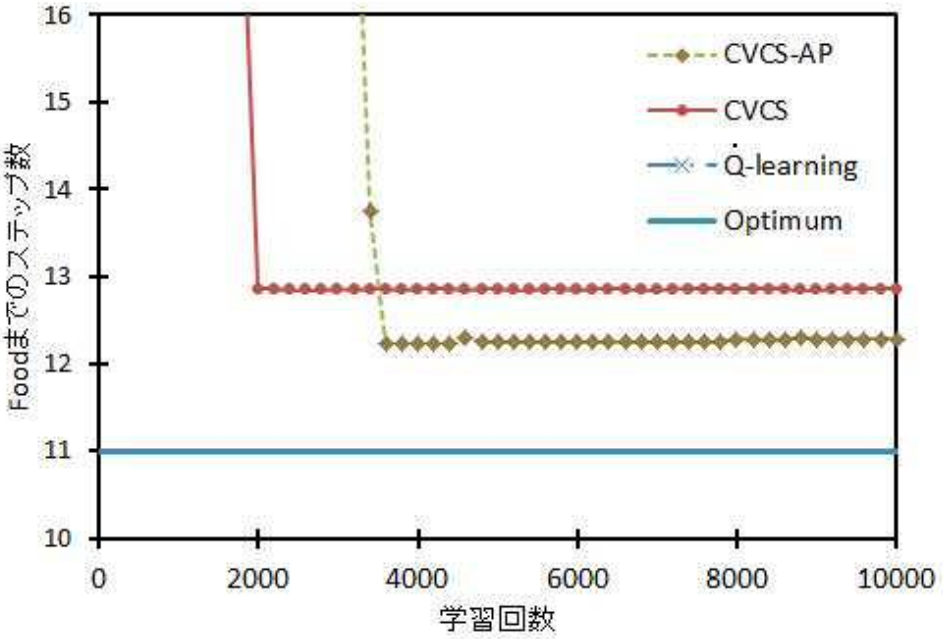

図 56　Type1-2 の学習結果（Optimum 付近の拡大）

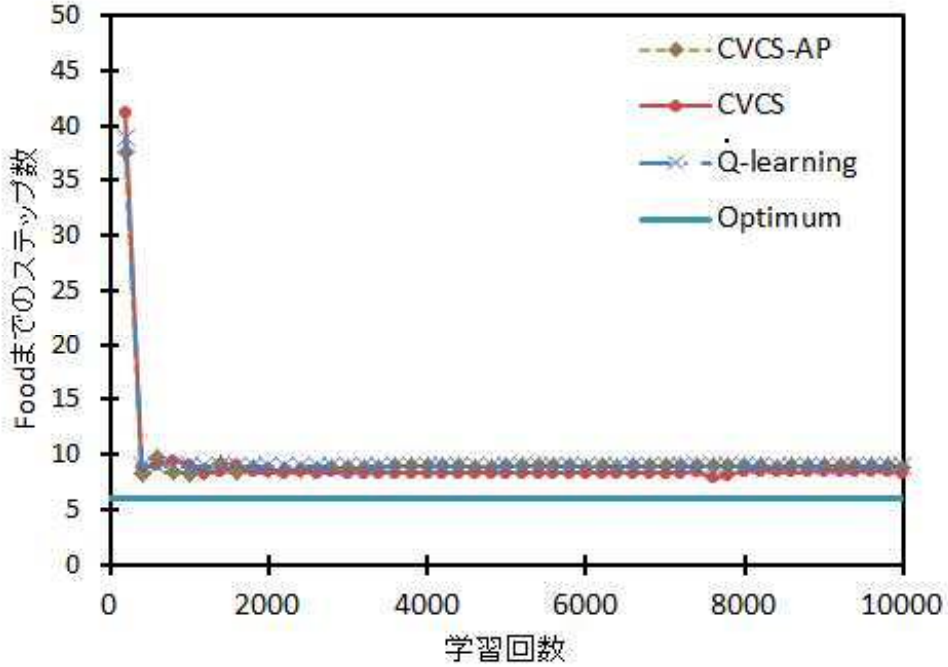

図 57　Type2’の学習結果

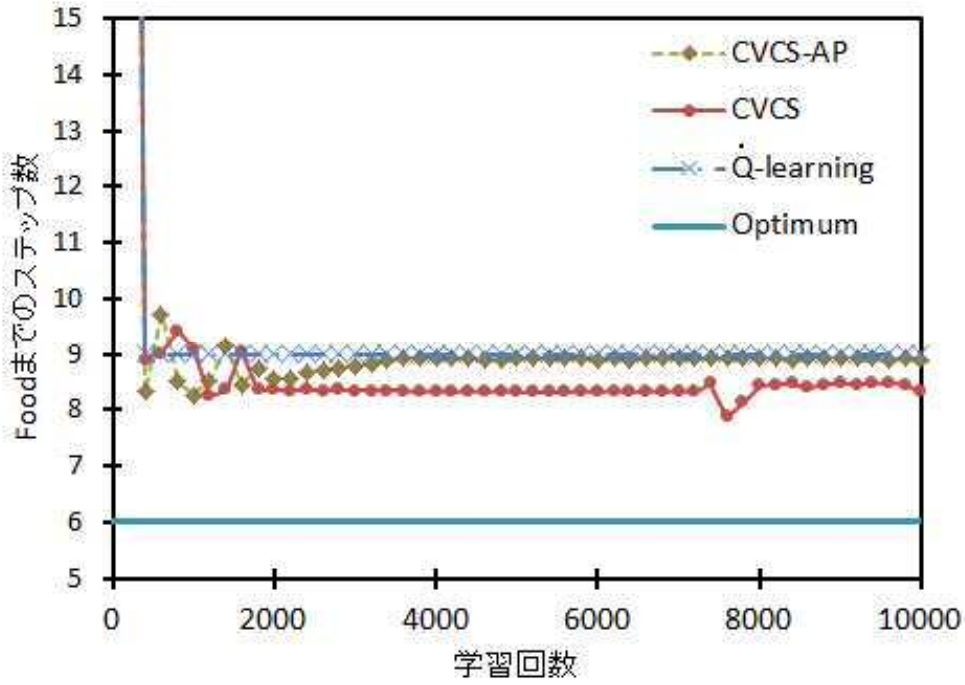

図 58　Type2’の学習結果（Optimum 付近の拡大）

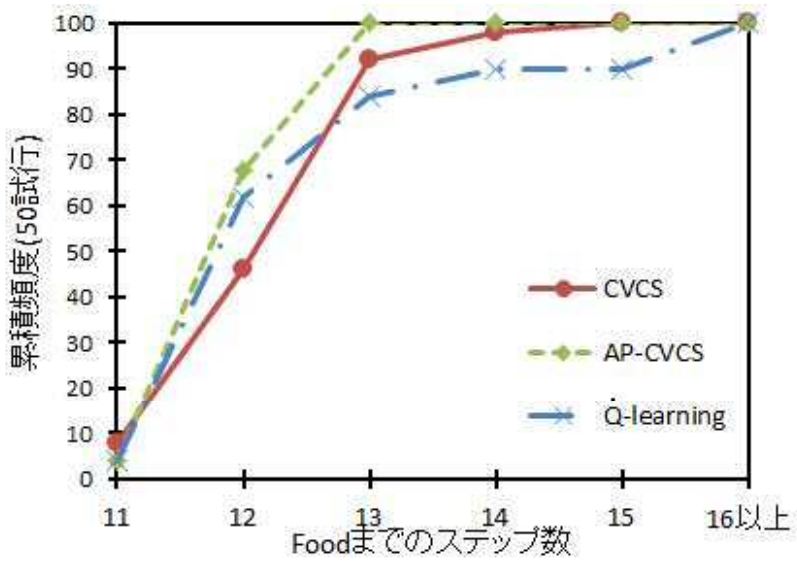

図 59　Type1-2 の学習結果の分布

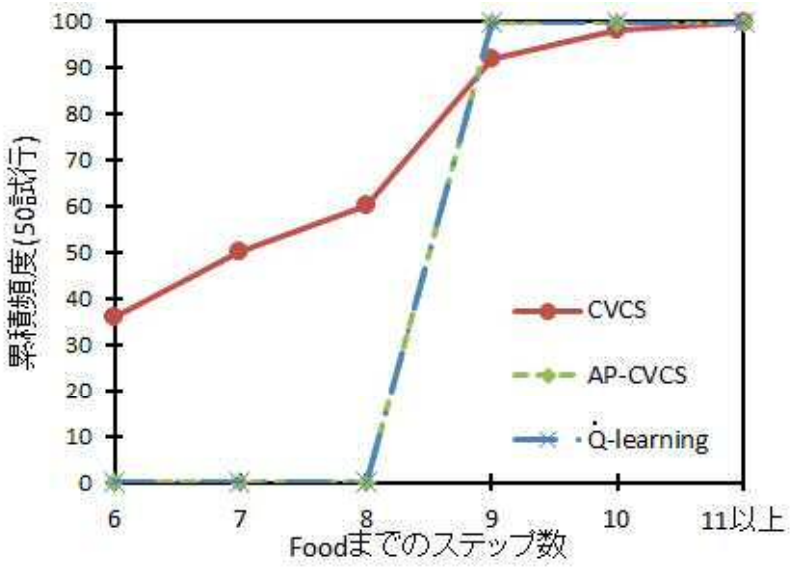

図 60　Type2'の学習結果の分布

8.3 考察

Type1-Large，Type2-Large および Type1-2 において，評価基準(1)について AP-CVCS が他の手法と比較して少ない平均ステップ数となっている理由については，5.4 節で述べた新たな淘汰の枠組によって，適切でない分類子の削除が促進されたためであると考えられる．これによって Tree の方向に移動するような不適切な方策の選択率を防ぐことで，多くの試行において最短ステップで Foods に到達するような方策を学習している．そのため，評価基準(2)でも AP-CVCS は Type1-Large および Type2-Large において最短ステップで Food に到達した試行の割合が他の手法と比較して非常に高い値となっている．また，Type1-2 では最短ステップで Food に到達した試行は他の手法と同程度ながら，全ての試行で最短ステップ+2 ステップ以内で Food に到達しており，7.1 節で述べたように適切な基本位相が設定できない環境でありながら，他の手法のように学習に失敗した試行は 0%となった．しかしながら，AP-CVCS は 5.5 節で述べたとおり，淘汰の促進を目的として様々な機構を導入している．この淘汰によって不適切な分類子が削除されるが，それと同時に本来は適切であるが学習が進んでいない分類子についても淘汰されるリスクが発生する．各手法の Type1-2 において最も報酬から離れた状態（初期状態）における適切な行動（左下に移動する行動）の強度値の分布を図 61，図 62 に示す．各図より，平均的に分布している CVCS と比較して AP-CVCS の強度の絶対値は 0 付近と 10 以上で 2 極化している．また，0 付近の値となった試行数は AP-CVCS の方が多い．これは促進された淘汰の枠組によって，ある程度小さな強度値を持つ個体が淘

汰されたためと考えられる．このように，学習が遅れる状態（報酬から離れた状態）の学習速度がより遅くなることで，Type1-2 において AP-CVCS の学習の遅れが確認されたと考えられる．

また，Type2’の評価基準(1)および(2)において AP-CVCS のステップ数が徐々に増加し，最終的に $\dot{Q}$–learning と同様の結果となった理由については，学習の過程で AP-CVCS の全分類子が他の分類子と異なるルールを持つことが原因である．CVCS および AP-CVCS における Type2’で獲得された各ルールの価値の絶対値および選択される初期内部参照値の例を図 63 に示す．7.5 節で述べたとおり，CVCS では Type2’において不完全知覚状態となる初期状態から発生する混同を，分類子数から分類子の強度の絶対値に差をつけることで知覚している．しかしながら，AP-CVCS では 5.4 節で述べた改良によって分類子集合の上限を減少させるとともに，他の分類子と重複しないルールを削除対象から除外している．そのため，分類子数の上限を最低限となる値まで減少させた場合，各ルールにおける分類子数は次第に 1 に近づくこととなる．これによって，AP-CVCS では Population の最大数の減少に伴って 7.5 節で述べた理由から最適解を獲得することができなくなり，最終的に CVCS と比較して劣る結果となってしまう．また，全ルールの分類子数が 1 つという状況は，$\dot{Q}$–learning における複素行動価値関数$\dot{Q}(s,a)$と実質的に同じであると考えられる．そのため，最終的に $\dot{Q}$–learning と同じ方策が獲得されていることから，$\dot{Q}$–learning と同様の収束結果となっている．しかしながら，図 58 に示すように Population の最大数が減少していない学習初期においては CVCS と同程度の性能を有し，$\dot{Q}$–learning と異なる結果となる．

最後に，第 8 章における実験結果のまとめとして，各環境における各手法の性能を示した表を表 6 に示す．表のマークは☆を最良として，◎，○，△，×の順に良好な結果であることを示している．表から分かる通り，提案手法は従来手法と比較してさまざまな環境で良好な結果を示しており，特に AP-CVCS は最適解の獲得率や学習の失敗率などの観点から広大な不完全知覚環境や Type1 と Type2 が混在する複雑な不完全知覚環境において CVCS よりも良好な結果を示した．AP-CVCS では他の手法と比較して評価基準(1)が同等ながらも評価基準(2)について良好な結果となったため，表 5 に示した各手法に対してより上位となる評価とした．

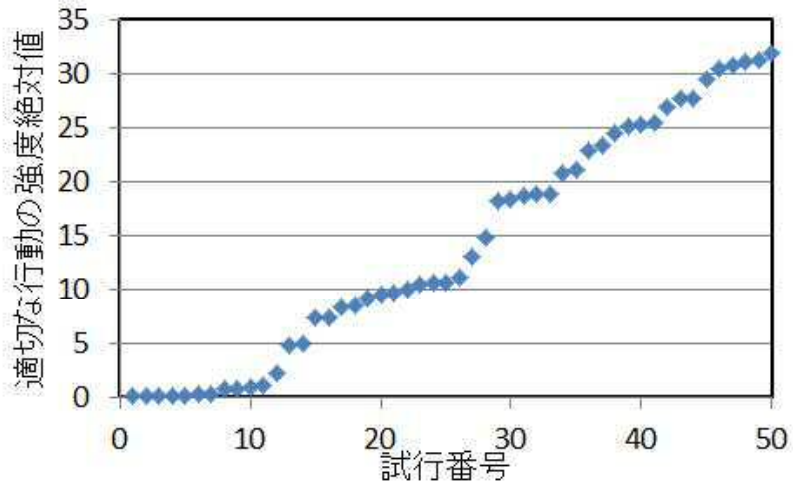

図 61　CVCS における適切な行動の強度絶対値の分布

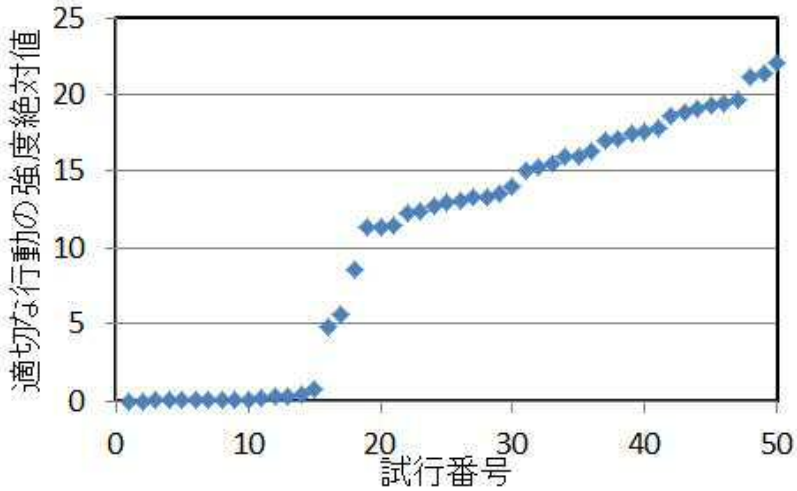

図 62　AP-CVCS における適切な行動の強度絶対値の分布

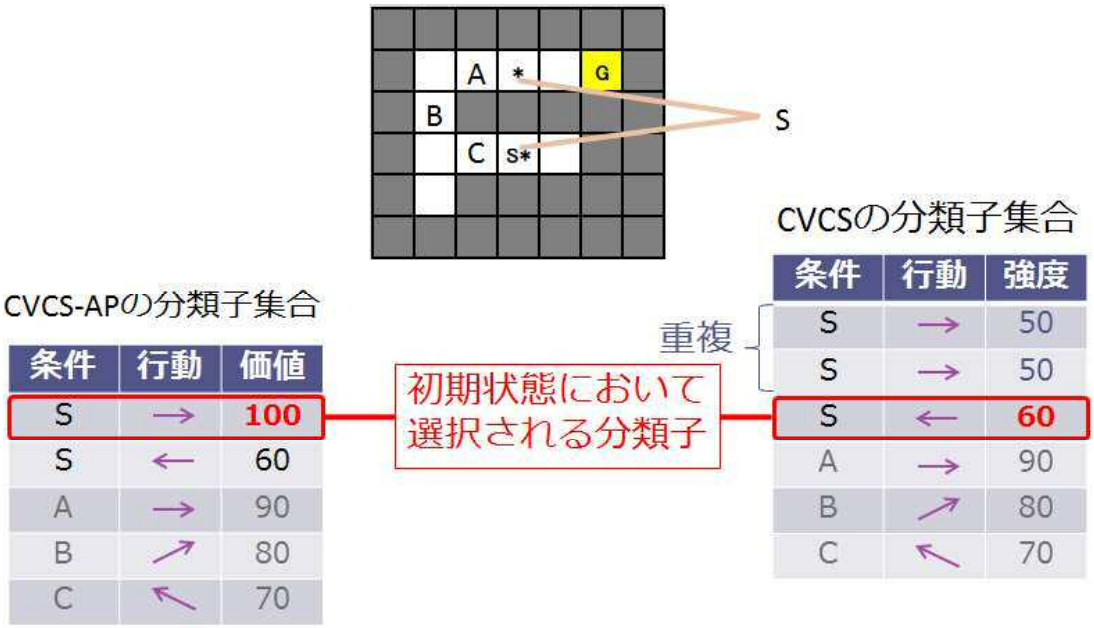

図 63　初期内部参照値の選択例

表 6　第 8 章における実験結果のまとめ

Environment	AP-CVCS	CVCS	$\dot{Q}$-learning
Type1-Large	☆	◎	○
Type2-Large	☆	◎	◎
Type1-2	☆	◎	△
Type2'	◎ 安定した学習	◎ 最適解を獲得可能	◎ 安定した学習

第 9 章　さらなる展開

本論文では提案手法によって，大規模・複雑な POMDPs 環境を効率的に探索し，従来手法と比較して高速に有効な解を獲得可能であることを示した．しかしながら，実環境に適用可能なアプローチに発展させるためには，いくつかの解決したい問題が考えられる．そこで本章では(1)動的な基本位相の設定法，(2)知覚入力に外乱の発生する環境下での学習の 2 点について述べる．

9.1 動的な基本位相の設定法

提案手法では，基本位相$\dot{\beta}$について，7.1 節で述べたように問題環境の特徴に合わせた値を静的に設定する必要性がある．しかしながら，問題環境（各不完全知覚間の Type やステップ数）が明らかでない環境に適用する場合，基本位相$\dot{\beta}$が適切に設定できないことから学習が困難となる．そこで本節では，基本位相$\dot{\beta}$の動的な設定機構を導入することで未知の環境に対しても適用可能な手法を目指す．

9.1.1　基本位相設定機構

基本位相が適切に設定され正しい学習が行われている場合は，学習時の試行錯誤的な行動による誤差はあるものの，5.3.2 節の式(13)，式(14)，式(15)で示した強度更新式によって強度の位相が変化することはない．すなわち，伝搬されてきた報酬や強度の位相が更新対象となる強度の位相と大きく異なる場合には基本位相が適切でないと判断することができる．

まず初めに，本機構を導入した際の基本位相$\dot{\beta}$の初期値$\dot{\beta}_0$を式(17)に示す．ここで|a|は学習エージェントが取りうる行動の種類数である．この設定によって，学習エージェントは基本位相が変化しなかった場合にも特定の状態において全ての行動が異なる位相を持つように強度を更新することが可能である．

$$\dot{\beta}_0 = \exp\left(\mathrm{i}\pi\frac{2}{|\mathrm{a}|}\right) \tag{17}$$

また, CVCS の強化部において強度の更新を行う前に, 以下に示す式(18), (19)の条件を満たす場合，基本位相が適切でないと判断して式(20)，(21)に示すように基本位相$\dot{\beta}$を再設定する．式(19)のδ_tは伝搬された報酬・強度と更新対象となる分類子集合の強度の位相差を表す．基本位相が再設定される条件は，伝搬された報酬・強度の位相と更新対象となる強度の位相差の絶対値が初期基本位相より大きい場合となる．また，式(21)のκ_tは不完全知覚間の推測ステップ数である．例として現在の基本位相が 45° である場合に，伝搬されてきた報酬が現在

の強度の位相と 180° 異なるケースを考える. この場合の不完全知覚間のステップ数は 180/45=4step と推測される. 推測したステップ数から計算される適切な基本位相と現在の基本位相を考慮して, 式(20)によって新たな基本位相が決定する. なお, 不完全知覚間の推測ステップ数κ_tが 2 以下となる場合には, その値に$2\pi/\arg\dot{\beta}$を足す. これは, 不完全知覚間において選択された強度の位相が一周以上回転している状況を想定している.

$$|\delta_t| > \dot{\beta}_0 \tag{18}$$

$$\delta_t = \arg\left[\left(r + \gamma \dot{S}^{(t)}{}_{max}\right) * \sum\nolimits_{cl_i \in [A_{t-1}]} \dot{S}_i\right] \tag{19}$$

$$\dot{\beta} \leftarrow \frac{\dot{\beta} + \exp(i\pi \cdot 2/\kappa_t)}{2} \tag{20}$$

$$\kappa_t = \frac{\delta_t}{\arg\dot{\beta}} \tag{21}$$

9.1.2 評価実験

基本位相設定機構を導入した提案手法の基本的特性の評価実験として, 不完全知覚問題として基本的なフィールドである Type2 の環境（図 24）に基本位相設定機構を導入した CVCS および AP-CVCS を適用した. Type2 のフィールド環境を問題環境に設定した理由としては, 本論文における他の環境では広大な状態空間や複雑な不完全知覚など, 学習を妨げる要因が多く, 基本位相設定機構の特徴が確認できない場合があるためである. パラメータ設定および評価基準に関しては 7.1 節に示したものと同様である.

各手法の学習結果を図 64 に示す. 各図において横軸は学習回数, 縦軸は学習終了時（Food 到達時）のステップ数である. 図 64 において緑, 赤のマーカー付の線はそれぞれ AP-CVCS, CVCS の結果を示し, 青の点線で示されるマーカーの線は AP-CVCS に, 青の実戦で示されるマーカーの線は CVCS に, それぞれ基本位相設定機構を導入した手法の結果である. 青の太い実線は Food までの最短ステップを示す. 図より, いずれの手法も基本位相を静的に設定した場合と比較すると性能が劣るもの, 学習に成功していることが確認できる.

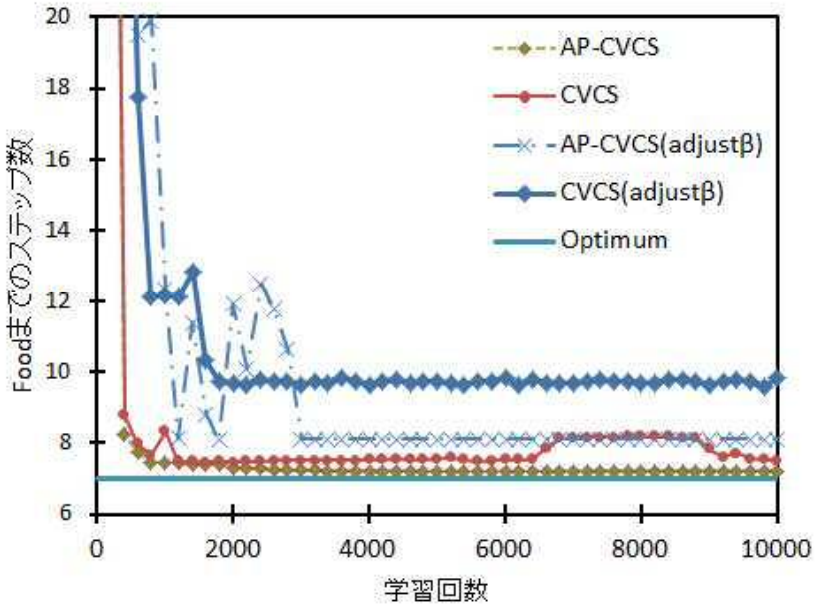

図 64　基本位相設定機構の学習結果

9.1.3　考察

各手法における 50 試行内での最終的な基本位相の値の分布を図 65，図 66 に示す．図から，両手法とも最終的な基本位相が 45° 前後と 120° 前後に多く分布していることが確認できる．Type2 のフィールドでは不完全知覚となる状態間を遷移するための最短行動数は 4step であることから，基本位相が 45° および 135° となった場合には不完全知覚となる状態における適切な 2 つの行動価値の位相差が 180° となる適切な位相となる．以上より，9.1 節で述べた機構によって適切な学習が可能な基本位相が設定可能であることが明らかとなった．しかしながら，いくつかの試行では適切な位相とはある程度異なる位相となっていることから，静的に基本位相を設定した場合と比較して Food までのステップ数が若干劣る結果となっている．CVCS に関しては 50 試行中 1 試行で最終的に学習が収束しなかったことから，平均的なステップ数を大きく引き上げている．学習に成功した試行のみの平均値では，AP-CVCS と同等の結果となっている．

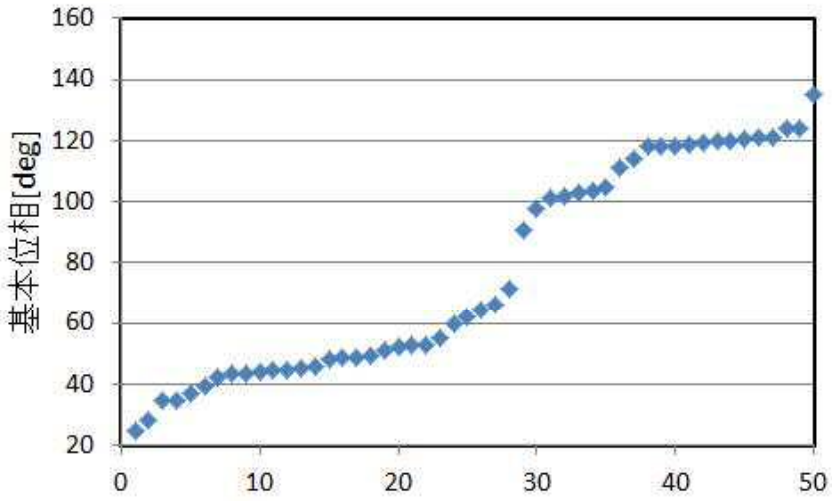

図 65　CVCS における最終的な基本位相の分布

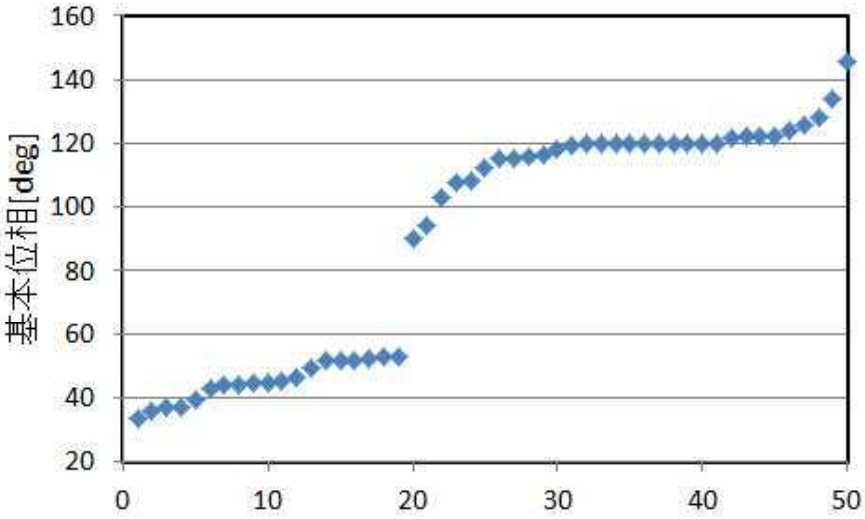

図 66　AP-CVCS における最終的な基本位相の分布

9.2 知覚入力に外乱の発生する環境下での学習

本論文では実問題によく確認される POMDPs 環境の中の不完全知覚問題に着目した手法を提案した．しかしながら，実問題においては学習エージェントの知覚入力に外乱（ノイズ）が発生するなどの理由で，次状態への遷移が確率的になる環境なども考えられる．そこで本節では，提案手法の知覚入力に外乱の発生する環境下での学習性能について調査する．

9.2.1 評価実験

外乱の発生する環境下における提案手法の基本的特性の評価実験として，知覚入力 1bit ごとに 1%の確率でランダムな bit が設定される Type2 のフィールド環境（図 24）に CVCS，AP-CVCS および $\dot{Q}$ –learning を適用した．パラメータ設定および評価基準に関しては 7.1 節に示したものと同様である．Type2 のフィールド環境を問題環境に設定した理由としては，9.1.1 節と同様に，本論文における他の環境では広大な状態空間や複雑な不完全知覚など，学習を妨げる要因が多く，外乱による学習への影響が確認できない場合があるためである．

各手法の学習結果を図 67，図 68，図 69，図 70 に示す．各図において横軸は学習回数，縦軸は学習終了時（Food 到達時）のステップ数である．図 67 において緑，赤，青のマーカー付の線はそれぞれ AP-CVCS，CVCS，$\dot{Q}$-Learning の結果を示し，青の太い実線は Food までの最短ステップを示す．また，図 68，図 69，図 70 では各手法について，ノイズのない環境との比較を表している．なお，ノイズによる影響を表現するため，移動平均でなく各学習時の結果を示している．各図において青の実線がノイズを有する環境，赤の点線がノイズのない環境での結果であり，緑の実線は Food までの最短ステップを示す．図 67 より，いずれの手法においても学習結果が安定していないことが確認できる．また，CVCS および $\dot{Q}$–learning と比較して AP-CVCS の学習結果が良好でないことが確認できる．外乱による学習結果への影響がない状態との学習性能の比較としては，CVCS では図 68 に示すようにノイズのない環境とほとんど変わらない結果となった．また，図 69 に示す $\dot{Q}$–learning では全体的に Food までのステップ数が 1step 強増加しており，結果としては若干の悪化が見られる．一方，図 70 に示す AP-CVCS では学習結果が一時的に損なわれる回数が他の手法に比べて多く，50 試行の平均としてはその他の手法と比べて最も悪い結果となった．

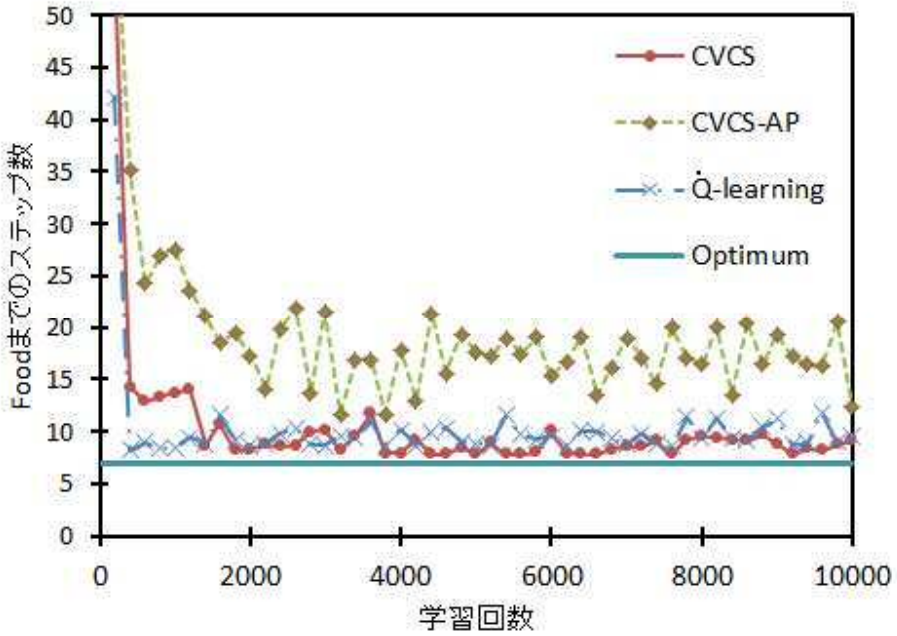

図 67　ノイズを有する Type2 の学習結果

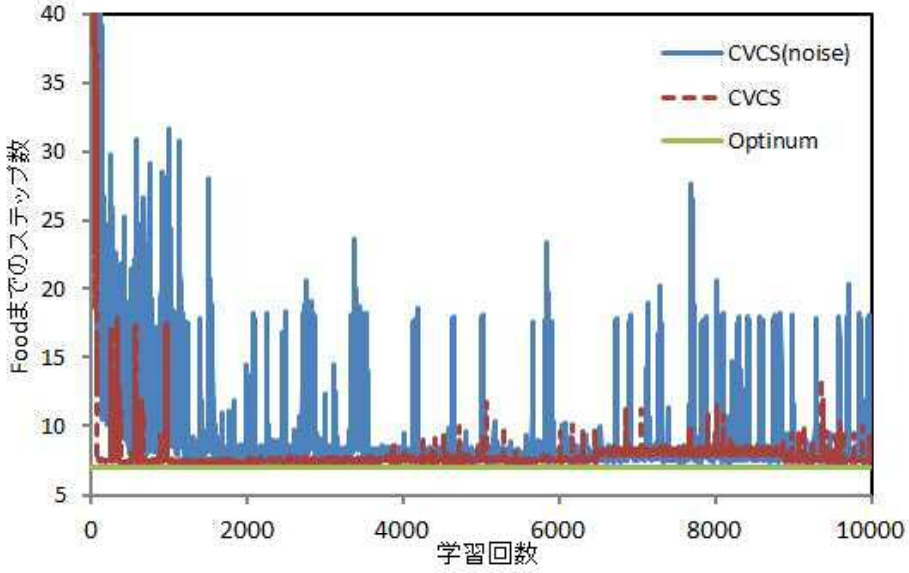

図 68　ノイズによる性能の比較（CVCS）

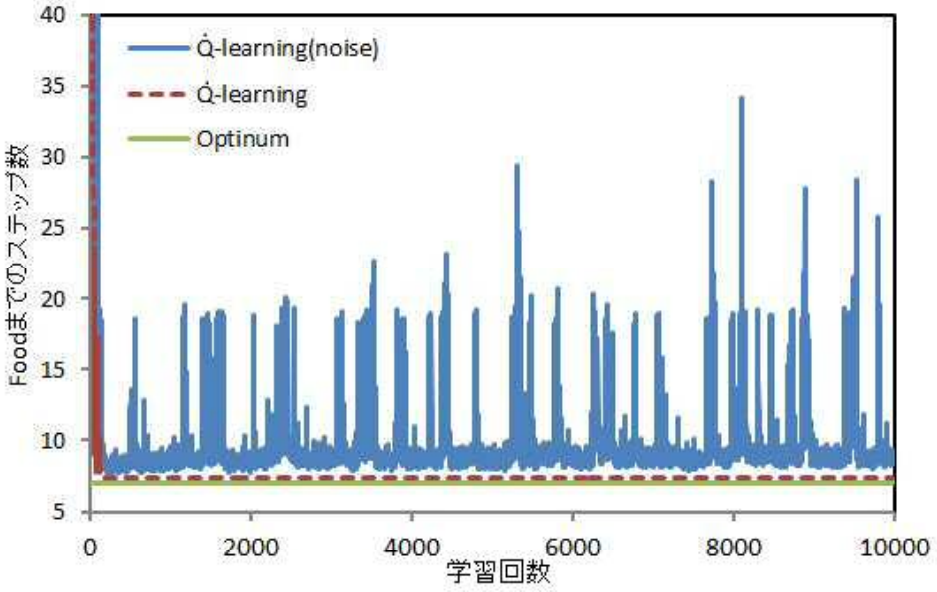

図 69　ノイズによる性能の比較（Q̇–learning）

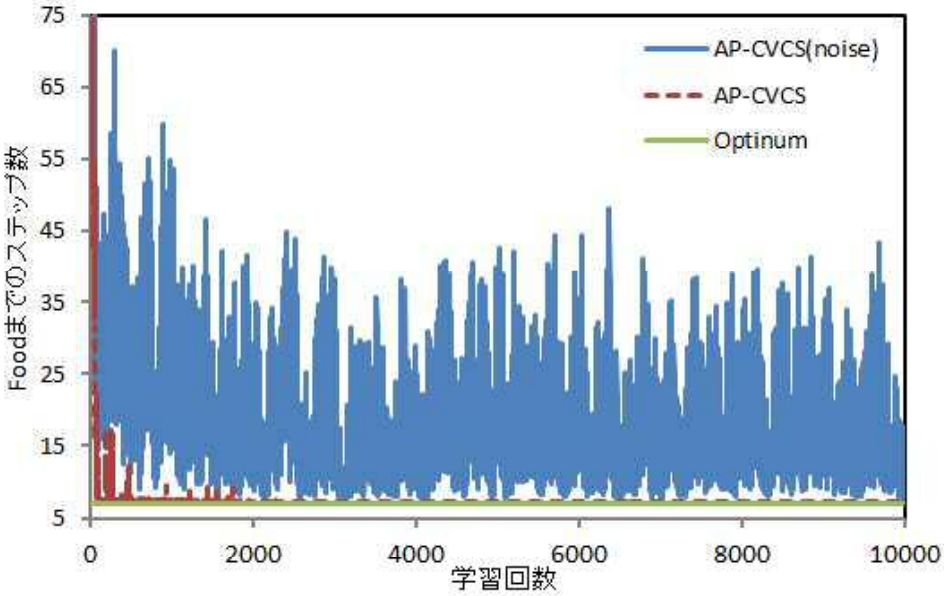

図 70　ノイズによる性能の比較（CVCS-AP）

9.2.1　考察

図 68，図 69，図 70 では各手法とも全試行で正しく学習している結果は確認されるものの，各試行の中で外乱の影響によって学習結果が一時的に損なわれていれる状況が観測された．また，手法によってその影響は大きく変化することとなった．Q̇–learning では得られた方策について 1step 強の悪化が確認で

きた一方，CVCS ではノイズのない環境とほとんど変わらない結果となった原因については，CVCS の持つ淘汰の枠組により，外乱によって発生した分類子を削除できたためと考えられる．また，淘汰の枠組を有するにも関わらず AP-CVCS では非常に不安定な学習結果となった理由については，AP-CVCS の持つ分類子上限数調整機構の影響であると考えられる. AP-CVCS では必要最低限まで Population の最大数が減少するため，外乱によって発生した分類子が Covering によって Population 内に生成された場合，その分だけ適切な状態における条件部を持つ分類子が排除されてしまうこととなる．そのため，外乱による分類子が発生した際に高確率で学習結果が破壊され，不安定な結果となったと考えられる.

第 10 章 結論

10.1 まとめ

本研究では, 複雑な POMDPs 環境下においても最適な方策を獲得するために, 複素強化学習に基づく学習分類子システム（Complex-Valued Reinforcement Learning-based Classifier System: CVCS）を提案した．提案手法は，行動履歴から不完全知覚状態を知覚し, 最適な方策を効率よく探索可能である．また，不要な分類子の削除を促進することで安定して最適な方策を獲得するため，分類子の最大数に着目して CVCS を改良した AP-CVCS を提案した．さらに，パラメータである基本位相の動的な設定機構を提案し，各手法に考案した．

提案手法の評価のため，異なる特性をもつ不完全知覚状態を有する Woods 問題について CVCS を適用したところ，次の知見を得た．まず，1）提案手法は，従来手法（$\dot{Q}$ -learning と ZCSM）よりも少ない学習回数で高い学習性能を実現した．具体的には，広大な状態行動空間を持つ環境において，各従来手法ではそれぞれの問題点から学習に多くの時間を必要としたが，提案手法ではそれらの問題点を改善することで高速な学習が可能となった．次に，2）従来手法では学習不可能な，不完全知覚に対して適切な基本位相が設定できない環境においても学習が可能であることを確認した．具体的には，Type1 と Type2 の不完全知覚が混在する環境において，ZCSM では環境の広さや複雑性の点で，$\dot{Q}$ -learning では Type1 と Type2 の両方に対して適切な基本位相が設定できないという点で学習が困難であったが，提案手法では各手法を組み合わせた枠組によって上記の問題を解決し，学習が可能となった．最後に，3）従来手法が最適な方策を獲得不可能な，初期状態が不完全知覚となる問題においても，提案手法は最適な方策を獲得可能であることを明らかにした．具体的には，提案手法では分類子の枠組を導入することで，初期内部参照値の決定法が報酬値によって受ける影響を削減することで，初期状態で適切な行動を選択することを可能とした．

また，同様の Woods 問題に対して AP-CVCS を適用したところ，1) CVCS や従来手法と比較してより安定した最適方策の獲得を達成した．具体的には，広大な状態空間を有する環境や Type1 と Type2 の不完全知覚が混在する環境において，Population の要素の最大数を減少させることで不要な分類子の淘汰を促進し，最適な方策のみを学習することが可能となった．一方，2）初期状態が不完全知覚状態となるような環境では従来手法と同等に安定した学習が可能であることを明らかにした．具体的には，最終的な各状態行動ルールの分類子数か

ら $\dot{Q}$ –learning と同様の方策が獲得されることとなるため，同様に安定した学習結果が得られることが判明した．加えて，パラメータに基本位相の動的な設定機構を CVCS および AP-CVCS に導入したところ，その機構によって事前に適切な基本位相が与えられていなくても正しい学習が可能となることが確認された．さらに，実環境における適用可能性の調査のために知覚入力に外乱を含む環境下に提案手法および従来手法を適用したところ，従来手法と比較して CVCS が環境の外乱に対して頑強性を有することが示された．

10.2 今後の課題

今後の課題としては，(1) MDPs 環境や Type3 に分類される不完全知覚環境，不完全知覚環境でない POMDPs 環境など，本論文で取り扱っていない問題領域への適用可能性の調査がある．また，(2) 現在の提案手法では進化計算法について淘汰のみしか考慮していないが，行動文脈を考慮した分類子の遺伝的操作法（交叉と突然変異）の考案によって，より効率的な方策の探索法を構築する．

参考文献

[1] Chrisman, L. : Reinforcement Learnng with perceptual aliasing: The Perceptual Distinctions Approach, Proc. of the 10th National Conference on Artificial Intelligence, pp.183-188, 1992.

[2] Cliff, D., Ross, S. : Adding Temporary Memory to ZCS, Adaptive Behavior, Vol. 3, No. 2, pp.101-150, 1995.

[3] Goldberg, D. E. : Genetic Algorithms in Search, Optimization, and Machine Learning, Addison-Wesley, 1989.

[4] Hamagami, T., Shibuya, T., and Shimada, S. : Complex-Valued Reinforcement Learning, Proc. IEEE International Conference on the Systems, Man and Cybernetics 2006, Vol. 5, pp.4175-4179, 2006.

[5] Holland, J.H.: Escaping brittleness: the Possibilities of General-purpose Learning Algorithms Applied to Parallel rule-based systems, in Michalski, R., Carbonell, J. and Mitchell, T., Eds., Machine Learning: An Artificial Intelligence Approach, Vol. 2, pp.593-623, Morgan Kaufmann, 1986.

[6] 井上 寛康, 高玉 圭樹, 下原 勝憲 : 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, 情報処理学会論文誌, Vol. 47, No. 5, pp.1483-1492, 2006.

[7] Jaakkola, T., Singth, S. P., Jordan, M. I.: Reinforcement Learning Algorithm for Partially Observable Markov Decision problems, Advances in Neural Information Processing Systems 7, pp.345-352, 1994.

[8] 蒋 励, 藤田 聡 : 小型乗合バスシステムにおける最適発車間隔問題のモデル化とその強化学習による獲得手法の提案, 情報処理学会研究報告. MPS 数理モデル化と問題解決研究報告, Vol.2003, No.20, pp.35-38, 2003.

[9] Kaelbling, L.P., Littman, M.L., and Cassandra, A.R. : Planning and acting in partially observable stochastic domains, Artificial Intelligence, Vol. 101, No. 1-2, pp.99-134, 1998.

[10] 木村 元, Kaelbling, L. P. : 部分観測マルコフ決定過程下での強化学習, 人工知能学会誌, vol. 12, No. 6, pp.822-830, 1997.

[11] 岸本 康秀, 滝口 哲也, 有木 康雄 : 階層的強化学習を適用した POMDP によるカーナビゲーションシステムの音声対話制御, 電子情報通信学会技術研究報告. SP, 音声, Vol.110, No.143, pp.49-54, 2010.

[12] Littman, M. L., Cassandra, A. R., Kaelbling, L. P.: Learning policies for partially observable environments, The 12th Intern. Conference on Machine Learning (1995)

[13] 宮本 行庸, 上原 邦昭 : 特徴構成法を用いた Q 学習の効率改善, 情報処理学会論文誌. 数理モデル化と応用, Vol. 40, No. SIG_9(TOM_2), pp.62-71 ,1999.

[14] 宮崎 和光, 荒井 幸代, 小林 重信 : POMDPs 環境下での決定的政策の学習, 人工知能学会誌, Vol. 14, No. 1, pp.148-156, 1999.

[15] 澁谷 長史, 濵上 知樹 : 複素数で表現された行動価値を用いる Q-learning, 電子情報通信学会論文誌, Vol. J91-D, No.5, pp.1286-1295, 2008.

[16] 下谷 篤史, 前田 新一, 石井 信 : 価値関数の分解による高速な強化学習法, 電子情報通信学会技術研究報告. NC, ニューロコンピューティング, Vol. 104, No. 759, pp.125-130, 2005.

[17] Stolzman, W.: An introduction to Anticipatory classifier system, Learning Classifier Systems, From Foundations to Applications, Springer, pp.175-194, 2000.

[18] Sutton, R., Barto, A.（著）, 三上 貞芳, 皆川 雅章（訳）: 強化学習, 森北出版, 2000.

[19] Watkins, C. J. C. H. : Technical note : Q-Learning, Machine Learning, Vol. 8, No. 3, pp. 279-292, 1992.

[20] Williams, M., Schmidhuber, J.: Solving POMDPs with Levin Search and EIRA, Proceedings of the 13th International Conference on Machine Learning, pp.534-542, 1996.

[21] Wilson, S. W. : ZCS: A zeroth level classifier system, Evolutionary Computation, Vol. 2, No. 1, pp.1-18, 1994.

[22] 吉見 隆洋, 田浦 俊春 : 階層型分類子システムを用いた視点制御過程の計算論的モデル, 計測自動制御学会論文集, Vol. 36, No. 10, pp.842-851, 2000.

[23] Zhanna, V. Z., Anthony J.B. : Learning Mazes with Aliasing States: An LCS Algorithm with Associative Perception, Adaptive Behavior, Vol. 17, No. 1, pp.28-57, 2009.

謝辞

本研究を行うにあたり，研究の進め方や発表練習，論文作成等について数多くの助言とご指導をいただきました髙玉圭樹教授に深く感謝致します．また，本論文の添削から研究室での生活まで，様々な面でお世話になりました中田雅也氏を始めとする髙玉研究室，佐藤研究室，服部研究室の皆様に深く感謝致します．最後に，指導教員として本論文のチェックをしていただいた吉浦裕教授に深く感謝致します．