知識を再利用するクラシファイアシステムの複数の環境への適応

井上寛康*(1)(2)　高玉圭樹(2)(3)　下原勝憲(1)(2)　片井修(1)
(1)京都大学大学院情報学研究科　京都市左京区吉田本町
(2)ATR 人間情報科学研究所　京都府相楽郡精華町光台 2 丁目 2 番地 2
(3)東京工業大学大学院総合理工学研究科　神奈川県横浜市緑区長津田町 4259 番地
e-mail: hirinoue@his.atr.co.jp

Abstract：本論文では，環境に対する既得の知識を再利用することを目指し，エージェントにとって完全に未知ではなく，既知の環境が混在した環境へ適応するシステムを提案した．　具体的には強化学習システムの 1 つであるクラシファイアシステムを複数使い分けるように拡張し，Woods シミュレーション環境でその有効性を確認した．

実験により，（1）ある 2 つの環境が混合された環境に対して，これら元の 2 つの環境の知識をあわせ持つクラシファイアを使うエージェントのほうが，新規のクラシファイアを持つエージェントよりもよい性能が得られること，および（2）ある 3 つの環境のうち任意の 2 つを混合した環境に対して，　3 つの環境のどの 2 つが混合の元になった環境かをエージェントが正しく識別することで，混合された環境に対応しない知識を含むクラシファイアを持つ場合や，新規のクラシファイアを持つ場合よりもよい性能が得られること，を確認した．

1　はじめに

海底や宇宙で活動するロボットやエージェントに自律性を持たせるための枠組みとして，強化学習が研究されている．このようなエージェントは，直接人間より指示を送ることが困難なため，予期しない事態にもできるだけ早く自律的に効果的な行動をとる必要がある．しかしながら，事前にエージェントに設計・学習させておいた知識で対処できない事態が発生した場合の対策は，これまで具体的に論じられてきていない．また，強化学習による適応は時間がかかるため，先に述べたエージェントが実際の環境で学習することは現実的ではない．　したがって，その時点でエージェントが持つ知識を再利用することが考えられる．

これまでにも，環境に対する知識を再利用する研究は行われている．（本論文で環境とは，エージェントの働きかける対象であり，エージェントに対してエージェントの置かれている状態を示すものとする．）１つは階層的強化学習[1]である．これは環境の中に含まれる，一連の手順により解決できる問題（サブタスク）をどういう順番で使うかを学習する．ところがこのサブタスクは人が作りこまねばならず，サブタスクが事前にわからない場合は，効果的に学習できず適応できない．もう１つは複数のクラシファイアシステムを環境に応じて使いわけるシステム[2]である．これは事前に学習した環境のうち，現在の環境がどれにあたるかを判別し，行動するというシステムである．しかしこのシステムは，学習済みの複数の環境を識別して適応することはできるが，未学習の環境へは適応できない．

そこで本論文では，エージェントが以前に活動していない環境であるが，エージェントにとって完全に未知ではなく既知の環境が混在している環境に適応するシステムを提案する．このような環境は，エージェント設計時に想定されている環境ではなく，また，何がサブタスクとなるかわからないよう

な環境である．簡単な例を示すと，エージェントが，環境 A では（青，赤，黄），環境 B では（緑，黄，青）という順番でブロックを操作することを学習しているとする．このとき，（青，赤，（緑，黄），黄）と A と B の混在した操作が必要であるような環境を考える，ということである．この例では環境 B の１部分が A と混在している．ただし，あまりにも小さい環境の断片が混在していると判断できないため，本論文ではある程度の大きさを持つ断片が混在しているとしている．このような環境について，既得の環境の知識を自律的に組み合わせ再利用するシステムを考える．ここで重要なのは，環境の断片が混在しているのであれば，それをサブタスクとして扱え，階層的強化学習が使えるのではないか，という疑問である．しかし，どのように環境が組み合わさるか事前にわからない環境を本論文では扱うため，サブタスクを事前に知ることはできず，階層的強化学習を使うことはできない．本論文ではファーストステップとして，我々の提案するシステムは，（１）すでに学習済みの２つの環境が混在している環境に対して，既得知識を組み合わせて適応できること，および，（２）すでに学習済みの３つの環境のうち２つが混在している環境に対して，自律的に正しい既得知識を組み合わせて適応できること，をシミュレーション実験で示す．

２　知識を再利用するシステム

本論文で提案するシステムはクラシファイアシステムを利用する．具体的にはエージェントが事前に学習する環境それぞれのために，複数のクラシファイアシステムを利用する．

２．１　クラシファイアシステム

クラシファイアシステムは，条件部と行動部を持つクラシファイアと呼ばれるルールの集合よりなる（図１）．クラシファイアは環境から得る報酬に基づいて強化され，目的のシステムを制御する[3]．本論文では，メッセージリストのないクラシファイアシステムである Zeroth level Classifier System (ZCS)[4]を用いている．これは，本論文の対象とする問題においてメッセージリストは主たる働きをしないためである．

強化学習システムの中でクラシファイアシステムを用いた理由は 2 つある．１つは，実装が容易であることである．もう１つは，ルールにより知識が表現されるので，可読性があるということである．すなわち，本論文が対象とする事前にサブタスクがわからない問題において，学習を終えたクラシファイアシステムを分析すると，サブタスクを発見でき，またサブタスクを解決する行動列を抽出できる．

２．２　実験したアーキテクチャ１

このアーキテクチャは，2 つの環境を事前に学習し，これら環境が混合された環境に適応するためのアーキテクチャである．このアーキテクチャには，事前に学習したどの環境の知識を組み合わせるのか，というアルゴリズムは入っていない．

このアーキテクチャは 3 つの ZCS を持つ．そのうちの 2 つは事前に学習するための ZCS であり，あと 1 つはこれら学習された 2 つの ZCS を組み合わせて生成される．ただし組み合わされる際には，強度の高い順でそれぞれの ZCS のクラシファイアを半分ずつ組み合わせる．クラシファイアが組み合わされる時，すでに持つ強度をどうするかという問題があるが，ここではリセットと何もしないの 2 通りについて実験した．

２．３　実験したアーキテクチャ２

このアーキテクチャは，3 つの環境を事前に学習し，これら環境のうち任意の 2 つが混合された環

境に対して適応するためのアーキテクチャである．このアーキテクチャは，事前に学習したどの環境の知識を組み合わせるべきかを自律的に調べるアルゴリズムを含んでいる．すなわち，事前に学習した 3 つの ZCS すべての組み合わせ，つまり 3 通りの ZCS を 2.1 節と同じ方法で作成し，どれが有効であるかを調べる．具体的には，これら 3 つの ZCS を環境と順にインタラクションさせ，もっとも報酬が得られる ZCS を採用し，その後のインタラクションを行う．

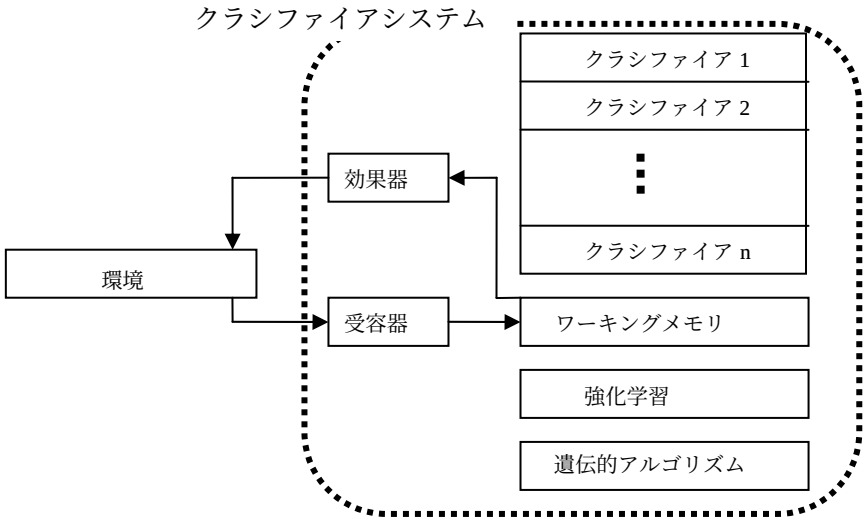

図 1：クラシファイアシステム

3　シミュレーション環境

3．1　Woods

Woods はマスの集合からなる 2 次元の世界である．Woods の各マスには障害(Obstacles)と空間，報酬(Food)が存在する（図 2）．報酬は 1 つの Woods に 1 つである．エージェントは Woods のマスの空間のいずれかに存在し，周囲 8 方向の情報を得ることができ，周囲 8 方向に進みうる．しかし，障害がある場合は進むことはできない．また，報酬がある場合は，報酬を取得し，Woods のいずれかの空間にランダムに移る．本論文では，1 つの環境が 1 つの Woods に対応する．

```
O O O O O O O O O
O . O O O O O . O
O . . O O O O . O
O . . . O O . O O
O . O O F ＊ O . O
O O O O O O . O O
O O O O . . . . O
O O O . O O O O O
O O O O O O O O O
```

O：　障害
．：　空間
F：　報酬
＊：　エージェント

図 2：　Woods の例

この Woods をシミュレーション環境として選んだのはいくつかの理由がある．

1. **実世界の環境のメタファ：**実世界の環境においてエージェントが問題を解決する際には，周囲の状況に応じて手順を実行する必要がある．Woods においても，エージェントは置かれた場所から報酬へ至る手順を実行する必要がある．したがって，Woods は実世界の環境のメタファとして捉

えられる．

2. **視覚的にわかりやすい：**シミュレーションの環境としてこれ以上視覚的にわかりやすいものは考えにくい．得られたシミュレーション結果の検討も容易である．
3. **従来の LCS との比較：**ZCS はこの Woods の環境において旧来の LCS との性能比較を広く行っている．すなわち，Woods においてエージェントに実装したシステムの性能を考察することは，すぐに従来の LCS との性能比較となる．

新規の Woods はランダムに生成される．ただし，どの空間からも必ず報酬に至る経路があるように，またマルコフ決定過程になるように生成される．

３．２　Woods の混合

本論文では，エージェントがこれまで学習した環境のうちの２つが混ざった環境に対して適応するシステムを提案するが，この混ざった環境は Woods を混合することで表現する．その手順は，Woods の上下・左右がつながっているものとして，Woods をランダムに 1 本の線で切断し，つなぎ合わせることにより実現する（図 3）．図の白の部分が混合の対象の部分を指す．ただし，Woods は離散的なマスからなるので，本当に直線で切断するのではなく，近似で行っている．

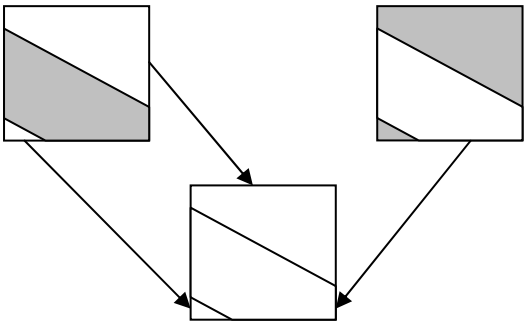

図３：　Woods の混合

このような Woods の混合は，「混合の元になった環境に存在している，事前には知ることができないサブタスクを持つ環境を作成すること」をあらわす．ここでサブタスクとは，報酬に至る経路の部分のことであり，混合の元になった Woods において一続きになっていた経路のことである．もし，任意に 2 つの Woods を混ぜるのであれば，ある程度の経路の長さを持つサブタスクは表現されない．ここで，ある程度の経路の長さ，というのは混合された Woods のすべての空間から報酬に至る最短経路の平均に対して，それら最短経路で通るサブタスクの経路の平均が，ほぼ半分程度になるということを示す．新規に Woods を作成するときと同様，混合 Woods についても，どの空間からも必ず報酬に至る経路があるように，またマルコフ決定過程になるように混合される．

４　シミュレーション実験

４．１　実験１：２つの Woods

Woods の設定

Woods は 2 つがランダムに生成され，これらの混合したものを 1 つ用意する．すなわち 3 つの Woods を用意する．Woods の大きさは，縦横 9 マスで外側は障害物である．また，それ以外の部分の障害物の生成確率は 0.7 であり，Woods の中心は常に報酬である．

エージェントの設定

2.2 節で解説したアーキテクチャを使用する．すなわち，エージェントは ZCS を 3 つ用意する．あらかじめ 2 つの ZCS で混合の元となる Woods をそれぞれ学習する．そして，これらの ZCS を混合し，3 つめの ZCS を作成する．ただし，この ZCS の混合の際に，強度をリセットする場合と，しない場合で別々に実験した．

ZCS のクラシファイアは条件部が 8 ビット，行動部が 3 ビットである．ZCS の各値の設定は，N（クラシファイアの数）=300，$P_{\#}$（条件部の don't care 生成確率）=0.33，S_0（初期強度）=20.0，β（学習率）=0.2，γ（割引率）=0.71，τ（税率）=0.1，χ（交叉の確率）=0.5，μ（突然変異の確率）=0.002，ρ（毎インタラクションの GA 発動確率）=0.25，φ（行動するクラシファイアが発動するかの閾値となる全体の平均の強度への係数）=0.5 である．

実験の説明

実験は，混合された Woods に対して，新規の ZCS を持つエージェント，事前に 2 つの ZCS で学習後，それらの ZCS のクラシファイアの強度をリセットして混合した ZCS を持つエージェント，同様の ZCS を強度をリセットしないで持つエージェントの 3 通りの実験を行った．

実験の結果

Woods とエージェントのインタラクションの 1 往復を 1 インタラクションとし，かつ 100 インタラクションを 1 試行とし，それぞれ 100 試行行った．結果を図４に示す．縦軸は 1 試行ごとにエージェントが得た報酬の数，横軸は試行数である．ただし，各プロットは 100 回の実験の平均である．

性能は強度リセットなしエージェント，強度リセットありエージェント，新規 ZCS エージェントの順によい．強度リセットなし・ありエージェントは，それぞれ第 1 回目の試行から多くの報酬を得ている． 学習は全てのエージェントで発生しており，3 つのグラフは近い値に収束している．

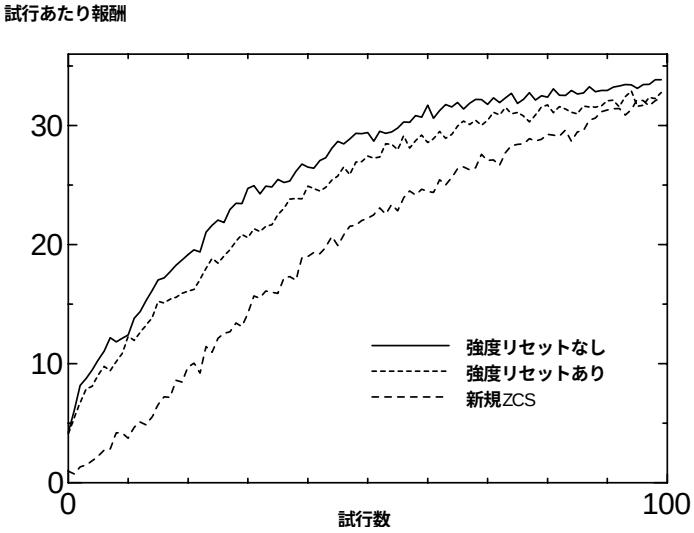

図４： 実験１の結果

４．２ 実験２：３つの Woods

Woods の設定

Woods は 3 つがランダムに生成され，これらのうちの 2 つを混合したものを 1 つ用意する．すなわち 4 つの Woods を用意する．各 Woods の仕様は実験 1 と同じである．

エージェントの設定

2.3 節で解説したアーキテクチャを使用する．すなわち，エージェントは ZCS を 6 つ用意する．あ

らかじめ 3 つの ZCS で混合の元となる Woods をそれぞれ学習する．そして，これら ZCS をすべての組み合わせで混合し，新たな 3 つの ZCS を作成する．ただし，この ZCS の混合の際に強度はリセットしていない．それぞれの混合した ZCS の，（混合された）Woods への再利用性を確認する試行回数は 1 回とした．つまり，混合した ZCS を順番に 1 つずつ選び，Woods と 1 試行の間インタラクションさせる．このとき，もっとも報酬を得られたものが選択され，インタラクションに用いられる．

ZCS の設定は実験 1 と同じである．

実験の説明

実験は，3 つある Woods のうちのある 2 つが混合された Woods に対して，新規の ZCS を持つエージェント，事前に 2 つの ZCS で学習後，混合した ZCS を持つエージェント（ただし，混合された ZCS は，混合された Woods に正しく対応していない．），事前に 3 つの ZCS で学習後，新たに 3 つの ZCS をこれらの 2 つずつ混合により作成し，選択するエージェントの 3 通りの実験を行った．

実験の結果

インタラクション，試行の定義，およびプロットの条件も実験 1 と同じである． 結果を図 5 に示す．

性能は，自律的に正しい混合 ZCS を選択するエージェント，誤った混合 ZCS を用いているエージェント，新規 ZCS の順によい．自律的に正しい混合 ZCS を選択するエージェントは，誤った混合 ZCS を用いているエージェントよりも，開始すぐ 2 倍程度多くの報酬を得ている．学習は全てのエージェントで発生し，3 つのグラフは近い値に収束している．

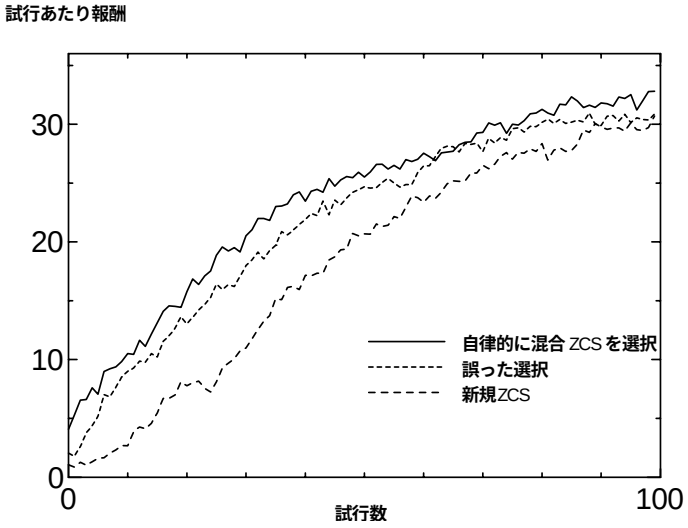

図５： 実験 2 の結果

5 論議

5．1 実験 1 の考察

混合された ZCS を持つエージェントは，新規の ZCS を持つエージェントよりもよい性能を示している．一方，強度をリセットしたエージェントはリセットしないエージェントよりも性能が悪い．

強度は学習の成果であるので，直感的には強度のリセットによる性能低下は当然と思われる．しかし，混合された環境にはエージェントにとって隠れたエイリアス（マルコフ決定過程をさまたげる状態）が存在するため，強度をリセットするほうが，リセットしないよりもより適応できるかの実験が必要と考えた．隠れたエイリアスとはすなわち次のようなものである．実験に使われた混合 Woods はマルコフ決定過程であるように作られているため，Woods の中にエイリアスはない．しかし，ある

混合の元となった Woods A と B では同じ状態のマスがありうる．ここで，混合 Woods にはその片方のみが含まれ，エージェントが混合 Woods 内のその状態に入ったとき，選択候補のクラシファイアが 2 つある．この際，強度がそのまま引き継がれており，かつ，強いほうのクラシファイアがエージェントを報酬に導かない場合，性能が低下すると予測した，ということである．しかしながら，このようなケースはまれであり，強度をリセットすることは前述のように優秀でないクラシファイアと同じ強度になるデメリットのほうが大きかったと考えられる．

2 つの環境が既知であるにもかかわらず，はじめから収束に近い値の性能を混合 ZCS が示さないことも重要である．これは以上の考察で述べたが，エージェントにとって隠れたエイリアスが存在するためと考えられる．エージェントはいくつかの状態において，この隠れたエイリアスにより性能低下を余儀なくされていると考えられる．すなわち，エージェントがある隠れたエイリアスの状態になった場合，とりうる行動が 2 つ示されることになり，そのどちらかを学習する必要が発生し，性能が低下すると考えられる．

５．２　実験 2 の考察

自律的に正しい混合 ZCS を選択しているエージェントは，誤った混合 ZCS を使っているエージェントに比べて，開始後すぐによい性能を示している．また同様に誤った知識を含む混合 ZCS は新規 ZCS よりよい性能を示している．この理由は，混合の元となった ZCS の片方は正しい知識であるためである．（元になる ZCS は 3 つしかなく，そのうちの 1 つだけが誤りであるため．）

５．３　問題点と今後の予定

本論文では，自律的に 2 つの既知の Woods の知識を再利用する実験を行った．この実験に用いたアーキテクチャ内には，既知の Woods のうちのどの 2 つが含まれるのかをテストするアルゴリズムが含まれていたが，このアルゴリズムには次のような問題がある．

- 既知の n 個の環境があると，このアルゴリズムでは ZCS の選択に $O(n^2)$の計算時間がかかってしまう．
- 混合した ZCS をテストする前に，既知の ZCS をそのまま使えるかをテストするべきである．これによりさらに ZCS の選択に計算時間がかかる．

このような問題を解決する糸口として，我々のこれまでの研究[5]を援用することが有効であると考える．この研究では，インタラクションの履歴を利用して，過去の知識の再利用性を並列に確認する手法について検討してきた．前述したが，これは事前に学習した環境のうち，現在の環境がどれにあたるかを判別し，行動するというシステムである．我々はこの手法を本論文のシステムに援用することで，上記の計算時間の問題を改善する考えである．

６　結論

本論文では，環境に対する既得の知識を再利用することを目指し，エージェントにとって完全に未知ではなく，既知の環境が混在した環境へ適応するシステムを提案した． 具体的には強化学習システムの 1 つであるクラシファイアシステムを複数使い分けるように拡張し，Woods シミュレーション環境でその有効性を確認した．

実験により，（１）ある 2 つの環境が混合された環境に対して，これら元の 2 つの環境の知識をあわせ持つクラシファイアを使うエージェントのほうが，新規のクラシファイアを持つエージェントよりもよい性能が得られること，および（２）ある 3 つの環境のうち任意の 2 つを混合した環境に対し

て，どの 2 つが混合の元になった環境かをエージェントが正しく識別することで，混合された環境に対応しない知識を含むクラシファイアを持つ場合や，新規のクラシファイアを持つ場合よりもよい性能が得られること，を確認した．

今後は，環境に関する知識を獲得する際にインタラクションの履歴を採取し，その履歴を元にして既知の環境のどの組み合わせが混在しているのかを認識するアルゴリズムを提案する．これにより本論文で提案したシステムで問題になった，既知の知識の最適な組み合わせの探索時間を削減できると考える．

謝辞

本研究は通信・放送機構の研究委託「人間情報コミュニケーションの研究開発」により実施したものである．

[1] R. Parr, and S. Russel: Reinforcement Learning with Hierarchies of Machines, Advances in Neural Information Processing Systems 10, pp.1043-1049 (1998)

[2] K. Takadama, H. Inoue, M. Okada, K. Shimohara, and O. Katai: Agent Architecture based on Interactive Self-Reflection Classifier System, International Journal of Artificial Life and Robotics (AROB), Springer-Verlag, to appear

[3] H. Holland: Properties of the Bucket Brigade Algorithm, The 1st International Conference on Genetic Algorithms (ICGA '85), pp.1-7 (1985)

[4] S. Wilson: ZCS: A Zeroth Level Classifier System, Evolutionaly Computation, 2(1), pp.1-18 (1994)

[5] 井上 寛康, 高玉 圭樹, 下原 勝憲: ユーザの内部状態の推測方法とそのエージェントアーキテクチャに関する考察, 第 20 回計測自動制御学会システム工学部会研究会「人工生命の新しい潮流」資料, pp.55-60 (2000)