# POMDP 環境下での強化学習と GA の融合手法の提案

A Hybrid Method of Reinforcement Learning and Genetic Algorithm for POMDP Environment

山城 啓秀 Yoshihide Yamashiro
上野 敦志 Atsushi Ueno
武田 英明 Hedeaki Takeda

奈良先端科学技術大学院大学 情報科学研究科
Graduate School of Information Science,
Nara Institute of Science and Technology (NAIST)

**Abstract**

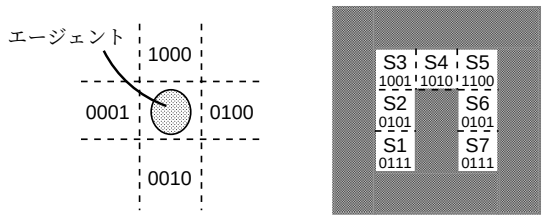
Reinforcement learning methods usually assume the environments in which the Markov property holds. But, an agent can not perceive states completely when the Markov property does not hold. An agent observes different states for the same ones when it causes perceptual aliasing. It is difficult for an agent to solve the task in such environment. This research proposes a new method called Delayed Reward-based Genetic Algorithm (DRGA) to solve POMDP (partially observable Markov decision problem) with perceptual aliasing. DRGA divides POMDP into multiple subtasks by decomposing an agent into multi subagents. Each agent learns appropriate policy series for fitting environment by using delayed reward. The policy is evolved with GA based on delayed reward obtained through the learning.

## 1 はじめに

強化学習はマルコフ決定過程 (Malkov decsion process, MDP) を対象とすることが多いが、エージェントが知覚できる情報は不十分なことが多い．このような知覚の見せかけ問題 (perceptual aliasing) などの問題が生じる部分観測マルコフ決定問題 (partially observable Markov decision problem, POMDP) 環境下ではエージェントが現状態を正確に知ることは不可能で、MDP を対象としたアルゴリズムでは十分な学習結果が得られなくなる．

しかし，エージェントはタスク遂行することに関しては，環境を完全に知る必要はなく，タスク達成のための環境モデルをエージェント内部に構築できれば良い．こういった観点から本研究はエージェントを複数のサブエージェントに分割し，知覚の見せかけ問題がタスク達成に影響を与える前に別のサブエージェントに制御を渡す．このような分割によって各サブエージェントは問題を MDP として扱うことができ、全体として POMDP を解決する．それぞれのサブタスクにサブゴールを設定し，分割されたサブエージェントはサブゴールに到達することを目的とする．サブゴールの必要性は，知覚の見せかけ問題のために一つのエージェントでタスクを解くのが困難なためであり，タスクをサブタスクに分解することで，タスクの難易度を下げる効果がある．このようにエージェントをサブゴールを持ったサブエージェントに分割することで，POMDP を解決する手法として**遅れ報酬に基づく遺伝的アルゴリズム** (Delayed Reward-based Genetic Algorithm, DRGA) を開発した．

POMDP を対象とした強化学習システムであ




各グリッドに壁がある場合の入力　エージェントが得る観測値

図 1: グリッド環境での知覚の見せかけ問題

る HQ-learning[5] との比較実験を通して DRGA が有効な手法であることを示す.

# 2 POMDP

## 2.1 知覚の見せかけ問題

知覚の見せかけ問題とは，不十分な知覚によって複数の異なる状態を同じ状態として知覚してしまうために起きる問題である.

グリッド環境を使用し知覚の見せかけ問題の例を紹介する．エージェントの知覚能力はグリッド環境で隣接したグリッドに壁があるかないかだけを知覚できるものとする．その場合，エージェントは図 1のような知覚入力を得る．図 1において，エージェントは S1 と S7 状態，S2 と S6 状態ではそれぞれ同じ知覚入力を得るため，どちらの状態であるかを知ることはできない．このように環境がマルコフ性でもエージェントの知覚能力によって POMDP を引き起こしてしまっている.

## 2.2 HQ-learning

HQ-learning は Q-learning[4] を階層的に拡張したアルゴリズムである．POMDP として問題とされている知覚の見せかけ問題を解くことを対象としている．部分観測をおこす問題で，サブゴールをおくことで，タスクをサブタスクに分割し，各サブタスクに対して Q-learning を用いて学習する枠組である．図 2に HQ-learning でのエージェントの構成図を示す.

エージェントは複数のサブエージェントによって構成され，サブエージェント 1 からサブエージェント m まで順番に制御が移って行く．制御が移るタイミングは各々のサブエージェントが自身のサブゴールに到達したときである．Qテー

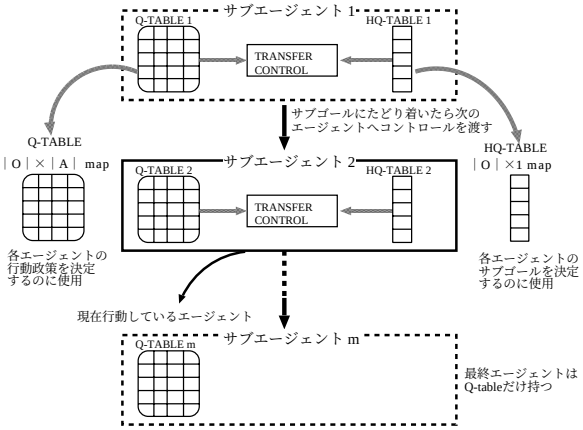



図 2: HQ-learninig － サブエージェント構成図

ブルは知覚と行動の組合せである $|O| \times |A|$ のQ値テーブルで，HQテーブルは各知覚状態のサブゴールとしての適切さの評価値で、$|O| \times 1$ のHQ値テーブルである．ここで，$|O|$ はエージェントが知覚できる情報の種類数，$|A|$ はエージェントが行える行動の種類数である．このようなエージェントの構成により，HQ-learning 異なるサブエージェントでの Q テーブルによる状態 $\rightarrow$ 行動を学習することで知覚の見せかけ問題を解決している.

# 3 DRGA

本章では，DRGA の実現方法について説明する．DRGA は HQ-learning のエージェント構成に着想を得，その欠点を補い，より効果的なアルゴリズムの設計を目指した.

## 3.1 表現方法

### 3.1.1 コード化 (encoding/decoding)

エージェントの知覚情報に対しての決定的な行動政策が GA においての遺伝子によって形成される．つまり，エージェントにおいて行動とは先天的行動であり，行動系列とその行動系列の目標 (サブゴール) を合わせてエージェントにおける政策と呼ぶことにする．以後，政策と表現するものは**政策：知覚集合 $\rightarrow$ 行動 + サブゴール情報**のことを指す.

一つの政策を 1 染色体として表現し，染色体にはサブゴール情報と知覚→行動が遺伝的に記

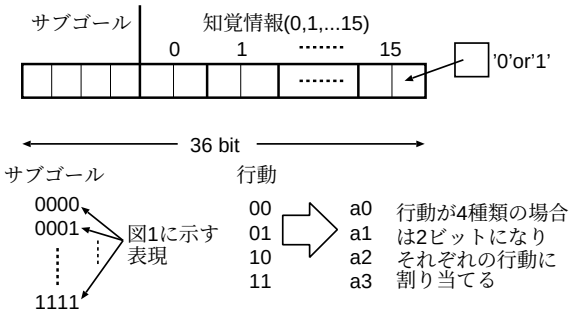
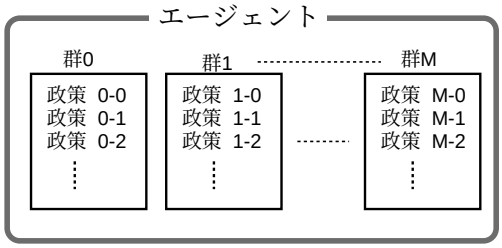



図 3: decording — genotype to phenotype




政策 m-i　m : 群番号
　　i : 政策番号

図 4: 群と集団

述されている．例えばエージェントの動作種類が $A$ 種で知覚できる知覚種類が $M$個であれば，1 遺伝子で $A^M \times M$種類の組み合わせを持つことになる．

GA ではコード化の設定が問題の解く性能を左右することが多い．そのため問題タスク依存でコード化をするのが望ましい．DRGA ではバイナリーコーディングを使用している．図 3でその手法について図示する．

### 3.1.2　エージェントと群

エージェント内部は複数の群によって分けられ，各群は複数の政策で構成される．政策は遺伝子によって表現されている染色体である．図 4 はサブエージェント＝群の説明を示したエージェントの構成図である．

各群は初期位置 (サブゴール到達時の初期知覚情報) を同じくする政策の集合である．HQ-learning アルゴリズムがはサブエージェントの順列固定のため，エージェントが観測した状態系列でサブゴール系列を学習する．このため，途中にサブゴールを置くと学習が破綻する．これを改善するために，DRGA ではサブエージェント (群) の形成は，ある特定のサブエージェントが動き出す場面 (知覚情報) を記憶すること (サブエージェントの初期知覚情報を固定) で環境により頑強に対応する群形成となっている．サブゴールの数は知覚情報の種類数と同じ数となるため，群数 $M$はサブゴールの種類数と同等数となる．すなわち，サブゴール到達時に，その到達したサブゴールの知覚情報によって次に行動するサブエージェントが決定され，選ばれた群内で政策を選別する．

### 3.1.3　進化手順

GA では一般的にまず個体が環境に対し行動などを起こし，環境からの評価として適応度を得る．この適応度をもとに集団に対して選択・淘汰が行われ，生き残った個体集団に対して交叉などの GA オペレータが施される．

DRGA では GA 処理 (選択・淘汰，GA オペレータ) は群内で行い，異なる群間での GA 処理は行わない．Goldberg[1] によると GA は buildingblock hypothesis によって集団 (複数点) を適応度の高い集団 (結果的に 1 点) にもっていく性質があるという．このため，全ての群を一緒にして GA 処理を行うとサブエージェントでなく一つのエージェントとなるため，一つのサブタスクしか解けなくなる．したがって異なる群間での GA 処理は行わない．GA 処理として行うのは以下のもので，全ての群で同様に行う．

- 選択 ‥‥‥ エリート+ルーレット戦略
- 交叉 ‥‥‥ 1〜3 点交叉
- 突然変異 ‥ 文字列変換 (ランダム)

通常 GA で用いる適応度関数と違い，DRGA で用いる適応度関数は遅れ報酬に基づいた評価値を使用して表現する (3.3節参照)．

## 3.2　遅れ報酬に基づく学習

各サブエージェントは遺伝的に保持している先天的行動から，適切な政策を選択するために後天的学習を行う．この後天的学習に Q($\lambda$)-learning[2][3] を使用する．

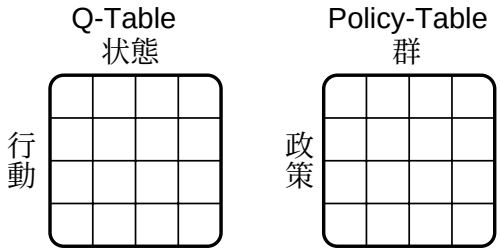


図 5: 学習テーブル表現

#### 3.2.1 政策選択

DRGA では，各サブエージェントにおいて政策選択に強化学習を用いる．一般的な強化学習は行動選択手法であるが，DRGA では政策の選択に用いる．DRGA で使用する強化学習アルゴリズムは Q($\lambda$)-learning である．DRGA での Q 値を **Policy – Table(政策テーブル)** と呼ぶと，図 5のように Q-Table と比較ができる．

Qテーブルでの状態は政策テーブルでは群を表し，Qテーブルで状態に対する行動とは，政策テーブルでは群に対する政策を表す．群内には複数の政策がありその政策種類がQテーブルでの状態内の行動種類に相当する．政策を選択した時，知覚 $\rightarrow$ 行動が決定的であるため，ある群で政策を選んだ時にサブゴールに到達するかどうかが暗に決定している．すなわち，ある群で任意の政策を選択したときの次の群は群の初期位置に対して決定的である．

強化学習を行う場合，ある程度の探索確率を残さなくてはならないため，行動の選択時に学習によって得た評価値によって各行動の選択確率を決定する．選択方策としては確率 $p$ で最大の評価値を持った行動を選び，確率 $(1-p)$ でランダムに選択する方法や，ボルツマン分布を使って行動を決定する方法など様々なものがある．DRGA で用いている方策はボルツマン分布に基づく選択である．

$$probability(p|g) = \frac{e^{Q(g,p)/T}}{\sum_{q \in P} e^{Q(g,p)/T}} \qquad (1)$$

$P$は群内の政策集合を表す．$g$は群，$p$ は群内の政策を示す．$probability(p|g)$ は群 $g$で，群内政策 $p$ が選ばれる確率を示す．$Q(g,p)$ は群 $g$における政策 $p$ の評価値であり，Q 値と記すことにする．$T$は政策を選ぶランダム率を調整する温度パラメータである．

#### 3.2.2 学習更新式

Q 値の更新には Q($\lambda$)-learning を用いる．時刻 $l$において群 $g_l$で政策 $p_l$を選択し，報酬 $r_l$が得られた時にシステムが Q 値を更新する式を以下の式 (2)，式 (3) に記す，ここで $l$ はサブエージェントの推移系列をあらわす離散時間である．サブエージェントのそれぞれで動作している時間は異なるが，サブエージェントの制御が移ることを 1 カウントとしている．

$$R_l^\lambda = r_l + \gamma[(1-\lambda) \max_{q \in P} Q(g_{l+1}, q) + \lambda R_{l+1}^\lambda] \quad (2)$$

$$Q(g_l, p_l) \leftarrow (1-\alpha) Q(g_l, p_l) + \alpha R_l^\lambda \qquad (3)$$

$\gamma$は割引率，$\alpha$は学習率で，共に $0< \gamma < 1$, $0< \alpha <1$ である．$R_l^\lambda$はエリジビリティを表す．$\lambda$はエリジビリティとユーティリティのトレードオフの割合で $0< \lambda <1$ である．

学習の更新は $L, L-1, .., 2, 1$ の順に計算する．$L$ はエージェントがタスクを行い，その終了条件 (タスク達成またはエージェントのタスクに対する寿命) を満たした時点での時刻である．式 (2) を計算し式 (3) に代入し，$Q(g_l, p_l)$ を更新する．この時，時刻 $L$ では $R_L^\lambda = r_L$ として計算する．

各政策はサブゴールに到達するか，タスクを達成するまで，もしくはエージェントの寿命が尽きるまで継続される．サブゴールに到達すると微小な報酬を得る．タスクを達成すると大きな報酬を得，その時の知覚状態に応じて次のサブエージェントに切りかわる．寿命が尽きると報酬は得られない．寿命はエージェントが持っており，1 政策で寿命を使い切るときもあれば，複数の政策で寿命を使い切るときもある．

### 3.3 DRGA 全体像

図 6は DRGA の全体像を示したものである．エージェントは環境との試行錯誤により得た経験に基づいて強化学習を行い，ある一定の試行数が終了するとエージェント自体が GA によって進化する．この時，GA においての個体表現は

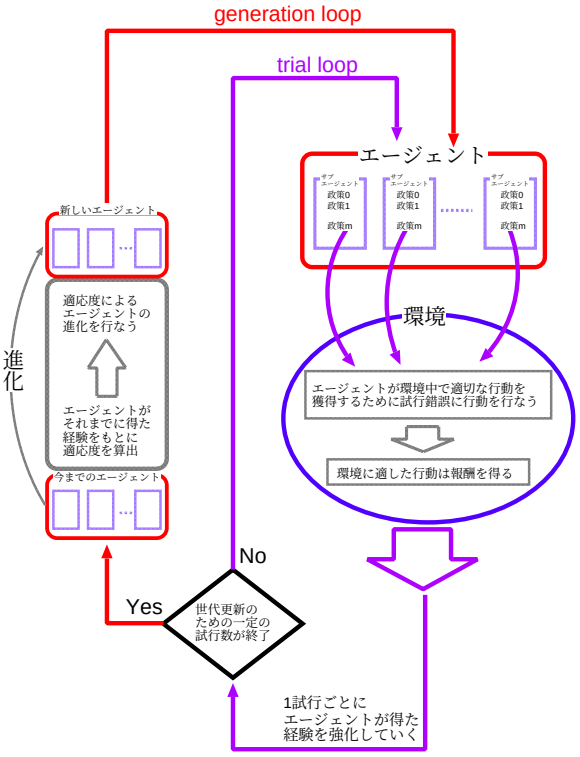


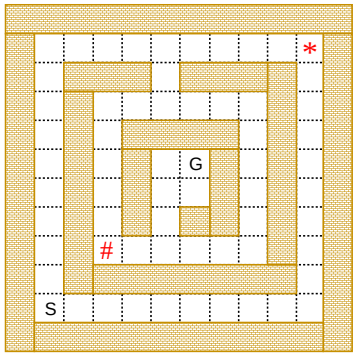

図 6: DRGA 全体像

DRGA での群内の政策にあたるため，進化の評価基準である適応度関数を図 5の政策テーブルの値で表現し GA を用いて進化を行う．進化後，今までの政策が変化するので政策テーブルの値を初期化する．そして，また新たな政策を保持したエージェントが環境に対して試行錯誤を繰り返しタスク達成を目指す．

# 4　シミュレーション実験

## 4.1　タスク － 10x10 迷路

まず，図 7に示す単純な部分観測迷路で実験を行った．この迷路は HQ-learning の著者が用意したもので，タスクは初期位置Ｓから目的地Ｇへの経路を発見することである．エージェントが行えるのは上下左右への 1 グリッド移動する 4 行動である．エージェントが得られる知覚情報は図 1に示したような隣接したグリッドに壁があるかないかだけである．この限定された知覚情報では得られる知覚種類が 16 種類しかない．そのためＳからＧの経路中に同じ観測値で違った行動をとらなくてはならない箇所が存在する．例えば，左右のみ壁である知覚入力では最初，上へ移動，




部分観測迷路で，ＳからＧまでの経路の走破．　この迷路での最短経路は 28 ステップ．＊，＃ は政策の再利用可能なサブゴールの位置

図 7: 10x10 迷路

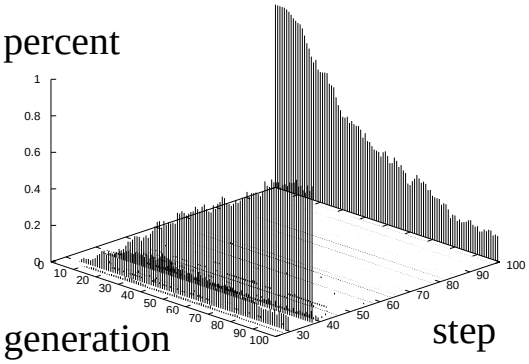



図 8: 10x10 迷路 実験結果 － ヒストグラム

次に下へ移動，最後に上へ移動しなければＧにはたどり着かない．そのため DRGA(HQ-learning も同様) では最低でも 2 回のサブエージェントの切り替えが必要となる．

## 4.2　結果 － 10x10 迷路

図 8は図 7の迷路を DRGA で解かせた結果である．各 generation 軸中の試行が何ステップでタスクを達成したかの割合を高さで示す．最終世代である 100 世代中では 38 ステップが一番多く試行の約 74%が 38 ステップを占めていることになる．タスクを達成出来なかったものはステップ 100 の約 13%であることを示している．

HQ-learning では，サブエージェントの順序系列に基づいて学習を行うため，サブエージェントが最低 3 つ必要である．DRGA では順序系列に基づいた学習ではなく，サブエージェントが推移する時，次のサブエージェントは知覚情報に応じて選ばれる．このため，同じ政策を再利用

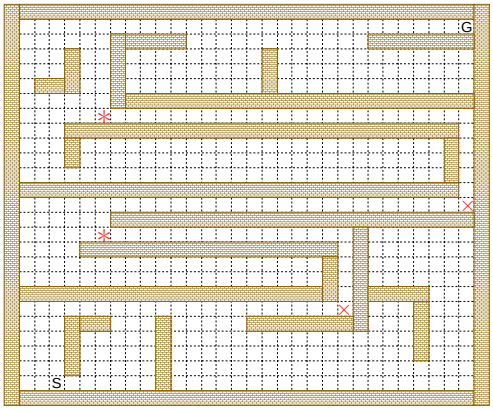



図 9: 30x25 迷路

することが可能である．例えば図 7の迷路であれば，最初に右へ移動して行き，1 つめのサブゴールを一番右端の’＊’に置く．次のサブゴールはＳと同じ知覚情報となる’＃’に置くと，最初に選択した政策を再利用しＧへたどり着くことが可能であり，サブエージェントが 2 つでゴールにたどり着くことが可能になる．この場合，群の中で競合が起こらないので，DRGA が見つけやすい経路となるためこの経路 (38 ステップ) の占有率が高い．

### 4.3 タスク － 30x25 迷路

次に，さらに難しい問題を用いて，DRGA と HQ-leraning の比較を行った．図 9に著者が作った迷路を示す．DRGA や HQ-learning にとっての難易度とはサブエージェントの切りかわり回数と経路の長さである．このタスクは知覚の見せかけ問題がタスク達成の妨げになる部分が 5 か所ある．すなわち，サブエージェントの切りかえを最低でも 4 回行わなくてはならない．図上の’＊’と’×’は DRGA が政策の再利用可能なサブゴールの位置である．

### 4.4 結果 － 30x25 迷路

比較実験の結果を図 10に示す．この図より DRGA はタスクを解いているが，HQ-learning はタスクを解くことに失敗しているといえる．100 世代目の実験結果ではタスクを解けないものが 18.8%に過ぎず，この値は世代更新すると適応度，Q値がクリアされることを考えると，十分な成績であると言える．

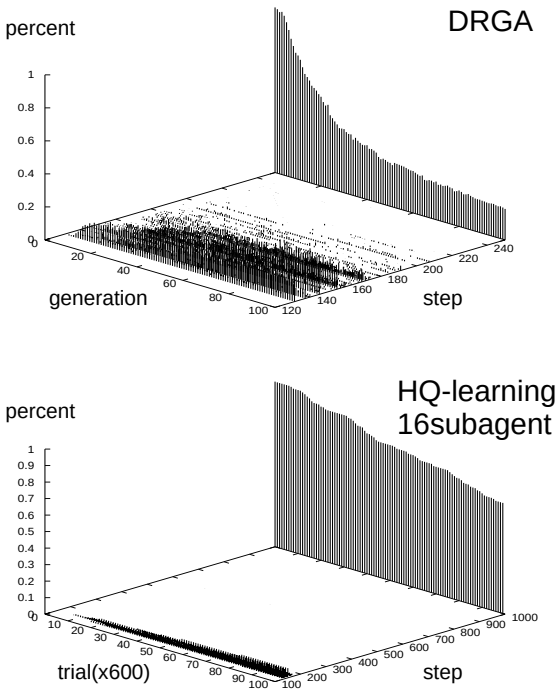



図 10: 30x25 迷路 実験結果 － ヒストグラム

## 5 おわりに

DRGA は HQ-learning と違い 1 ステップごとの行動にランダム性が含まれず，決定的に行動するため，長い経路でも効率良くタスクをサブタスクに分割することができれば，十分に POMDP を解決できることが分かった．また，DRGA は政策の再利用を行うことで，タスクの難易度を下げ HQ-learning よりも良い結果を残すことができた．

## 参考文献

[1] Goldberg, D.E. :Genetic Algorithms in Search, Optimizetion, and Machine Learning, *Addison-Wesley*, 1989.

[2] Peng, J. :Efficient Memory-Based Dynamic Programing *Proceedings of the 12th International Conference on Machine Learning*, pp.438-446 , 1995.

[3] Peng, J. and Williams, R. :Incremental multi-step Q-learning, *Machine Learning*, 22, pp.283-290 , 1996.

[4] Watkins, C.J.C.H.,and Dayan, P. :Q-learning, *Machine Learninig*, 8, pp.279-292 , 1992.

[5] Wiering, M. and Schmidhuber, J. :HQ-Learning, *Adaptive Behavior*, Vol. 6, No. 2 , 1997.