論　文

遅れ報酬に基づく遺伝的アルゴリズムによる部分観測マルコフ決定問題の解決手法

山城　啓秀[†*]　　上野　敦志[†]　　武田　英明[††**]

Delayed Reward-based Genetic Algorithms for Partially Observable Markov Decision Problems

Yoshihide YAMASHIRO[†*], Atsushi UENO[†], and Hideaki TAKEDA[††**]

あらまし　強化学習は通常マルコフ性が仮定されていることが多い．しかし，エージェントが環境を完全に観測できるとは限らず，そのような場合，異なる状態を同一の状態として観測する．本研究ではこのような知覚の見せかけ問題を伴う部分観測マルコフ決定問題（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（DRGA）を開発した．DRGA は POMDP を複数のサブタスクに分割し，エージェントを複数のサブエージェントに分割することで POMDP を解決する．それぞれのサブエージェントは遅れ報酬を用いることで環境に適した政策を獲得し，政策自体は学習を通して得られた遅れ報酬に基づき遺伝的アルゴリズムによって進化する．淘汰されて生き残った有効な政策を組み合わせることによって環境に適応する．本手法を知覚が限定された迷路探索問題に適用し，その有効性を示す．

キーワード　強化学習，遺伝的アルゴリズム，部分観測マルコフ決定問題，知覚の見せかけ問題

1. まえがき

強化学習はマルコフ決定過程（Markov decision 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．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という問題が起こる．これを知覚の見せかけ問題（perceptual aliasing）[1] という．このような知覚の見せかけ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を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 を解決することができる．タスクの分割は，各サブタスクにサブゴールを設定することで行う．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．問題は，タスクをうまく分割するためのサブゴールの学習と，各サブエージェントにおける MDP の学習である二つのレベルに分けられる．

このような不完全知覚の問題では，知覚が不完全なためにかえってうまく抽象化することができて，色々な場面で同じ政策が共通して通用するということがあり得る．異なる地点が同じように見える場合，それぞれで違う行動を選択する必要がある場合にはタスク達成が困難になるが，どちらも同じ行動を選択すれば良い場合には，知識をうまく再利用することできるのである．そこで，本論文では，このような不完全知

† 奈良先端科学技術大学院大学 情報科学研究科，生駒市
Graduate School of Information Science, Nara Institute of Science and Technology (NAIST)
†† 国立情報学研究所，東京都
National Institute of Informatics (NII)
* 現在，(株) 三洋電機
** 併任　奈良先端科学技術大学院大学

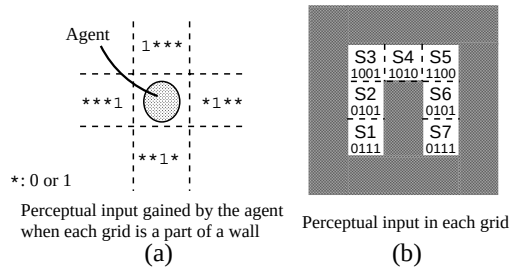


図 1　グリッド環境での知覚の見せかけ問題
Fig. 1　Perceptual aliasing in grid space.

覚の大規模問題において色々な場面で同一の小問題がたくさん現れるような問題を対象として，**遅れ報酬に基づく遺伝的アルゴリズム**（Delayed Reward-based Genetic Algorithm for POMDP, DRGA）を提案する．DRGA では，タスク全体を見て問題を MDP に分割しながら，色々な場面で役に立つ政策が生き残り，再利用可能な場面を見つけてそれらの有効な政策を適用していくことによって大規模な部分観測マルコフ決定問題に対応する．

このように POMDP をサブタスクに分割して解く先行研究として HQ-learning [2] が提案されている．しかし，HQ-learning では個々のサブエージェントが全く独立に学習を行っていて，有効な政策の再利用は全く考えていない．同一の小問題に分けることができるような不完全知覚の大規模問題を扱う場合，全く独立に学習するのは非効率的である．本稿では HQ-learning との比較実験を通して DRGA の有効性を示す．

2. POMDP

2.1 グリッド環境における知覚の見せかけ問題

本研究では，次のようなグリッド環境における知覚の見せかけ問題を扱う．エージェントの知覚能力はグリッド環境において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．すなわち，エージェントの知覚入力は４ビットで表され，先頭のビットから順に，１である時，それぞれ北東南西方向のグリッドに壁があることを示すものとする（図 1a）．図 1b において，エージェントは S1 と S7 の位置，S2 と S6 の位置ではそれぞれ同じ知覚入力を得るため，どちらの位置であるかを知ることはできない．この問題では，環境はマルコフ的であるがエージェントの知覚能力の不足によって POMDP と見なせる．

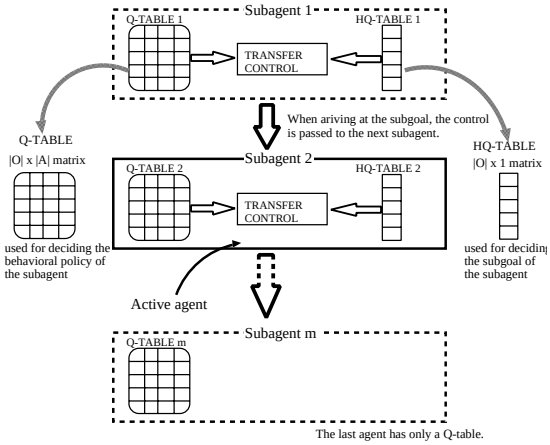


図 2　HQ-learning － サブエージェント構成図
Fig. 2　Agent architecture of HQ-learning.

2.2 HQ-learning

HQ-learning は Q-learning [3] 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．部分観測問題において，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される．m はあらかじめ決められた数である．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時である．サブゴールは知覚状態の内の一つとして定義される．

各サブエージェントは Q テーブルに基づいて Max-Boltzmann ルールで行動を決定する．Q テーブルは Q 値（行為価値）のテーブルであり，知覚状態集合 O，行動集合 A とすると，知覚と行動の組合せである $|O| \times |A|$ の大きさを持つ．Max-Boltzmann ルールは確率 P_{max} で最大 Q 値の行動を選択し，確率 $1 - P_{max}$ で次の式 (1) で表されるボルツマン分布に基づく確率で行動を選択する．

$$prob_o(a) = \frac{e^{Q(o,a)/T}}{\sum_{a' \in A} e^{Q(o,a')/T}} \qquad (1)$$

$prob_o(a)$ は知覚状態 o で行動 a を選択する確率である．T は行動を選ぶ際のランダム性を調整する温度パラメータである．

サブゴールの学習は HQ テーブルを用いて行う．

HQ テーブルは HQ 値（各知覚状態のサブゴールとしての適切さの評価値）のテーブルであり，$|O| \times 1$ の大きさを持つ．サブゴールは HQ テーブルに基づいて Max-Random ルールで決定される．すなわち，確率 P_{max}で最大 HQ 値の知覚状態を選択し，確率 $1 - P_{max}$でランダムに選ぶ．

HQ-learning では，各状態遷移を記憶しておき，エージェントがタスク終了時に一括して Q 値，HQ 値の更新を行う（オフライン更新）．Q 値は offline Q(λ)-learning [4] を用いて更新され，HQ 値は以下の式 (2) で更新される．

$$R_i = \sum_{t=t_i}^{t_{i+1}-1} \gamma^{t-t_i} R(s_t, a_t)$$

$$HQ'_i(\hat{o}_i) \leftarrow R_i + \gamma^{t_{i+1}-t_i}[(1-\lambda) \max_{o' \in O} HQ_{i+1}(o') + \lambda HQ'_{i+1}(\hat{o}_{i+1})]$$

$$HQ_i(\hat{o}_i) \leftarrow (1-\alpha_{HQ}) HQ_i(\hat{o}_i) + \alpha_{HQ} HQ'_i(\hat{o}_i) \quad (2)$$

ここで γ $(0 \leqq \gamma \leqq 1)$ は報酬の割引率であり，将来報酬と即時報酬のトレードオフの割合を示している．$R(s,a)$ は状態 s で行動 a をとった時に得られる報酬である．従って，R_iはサブエージェント i の動作している時刻 t_iから $t_{i+1} - 1$ に得た報酬の総計を示している．$HQ_i(o)$ はサブエージェント i の知覚状態 o に対する HQ 値，$\hat{o}_i$はサブエージェント i でその時選ばれたサブゴール，α_{HQ} $(0 \leqq \alpha_{HQ} \leqq 1)$ は HQ テーブルの学習率である．$HQ'_i(o)$ は eligibility traces(注 1)を用いた場合の目指すべき HQ 値であり，λ $(0 \leqq \lambda \leqq 1)$ は eligibility traces を用いる程度を表す定数である．

2.3 HQ-learning の問題点

HQ-learning は，問題を MDP で表されるところまで分割し，決められた順番通りにサブエージェントを適用していく．このようにエージェントの内部変数のみに基づいてサブエージェントを決定する手法では，サブエージェントを切り替える地点でどの政策が有効かの指標が全くないので，既知の政策を手当り次第に試してみるか，全く独立に学習し直すしか方法がない．HQ-learning では後者の方法を用いていて，有効な政策の再利用は全く考えていない．本論文で扱うような同一の小問題に分けることができる不完全知覚の大規模問題を扱う場合，全く独立に学習するのは非効率的である．そのため，ゴールに到達するまでに多くのサブエージェントが必要な時には環境に適応できない場合がある．HQ-learning の元論文 [2] で扱っている問題は，最も難しい問題でも最小で 3 サブエージェントで解けてしまうので，もっと大規模な問題に対する適用性は検証の余地がある．

(注 1)：eligibility traces とは，状態における評価値（Q 値や HQ 値）を計算する時に，状態の遷移系列上の直後の 状態の評価値のみを考慮するのではなく，それ以降の状態の評価値も考慮に 入れて計算するという概念であって，強化学習の報酬の分配を効率良くした り，部分的にマルコフ性が成り立たない（non-Markov）問題に対応したりす る効果がある．詳しくは文献 [5] 参照．

3. 本研究の目的と手段

本研究では，部分観測マルコフ決定問題において，色々な場面で同一の小問題がたくさん現れるような問題を対象とする．そして，タスク全体を見て問題を MDP に分割しながら，知覚的見せかけを利用して色々な場面で同じ政策を再利用することによって，大規模な部分観測マルコフ決定問題に対応できる手法の開発を目的とする．

知覚の見せかけ問題を対象とする手法があえて知覚的見せかけを残すのは不自然に思われるかも知れない．しかし，異なる地点が同じように見える場合，それぞれで違う行動を選択する必要がある場合には困った問題を引き起こすが，どちらも同じ行動を選択すれば良い場合には，知識をうまく再利用することができる．従って，政策を再利用しようとする学習システムにとって，サブエージェントの切替え地点で知覚的見せかけが起こるのは本質的な特徴である．本研究では，タスク達成にとって障害となる知覚的見せかけを排除し，障害とならない場合は逆にうまく利用して効率良く問題を解くことを目指す．

そのための手段として，サブエージェントの切替え地点の知覚情報に基づくサブエージェントの選択を提案する．これは，例えば「前に壁がある」という知覚情報に基づいて「壁に沿って歩く」という政策を選択するといった人間の政策選択法からの類推である．色々な場面で同一の小問題がたくさん現れるような問題を扱う場合，ある政策がどの場面で有効であるかは環境に依存している．切替え地点の知覚情報に基づいてサブエージェントを選択することによって，環境に応じたサブエージェントの切替えが期待できる．

4. DRGA

本章では，DRGA アルゴリズムの全体像と特徴を示し，その実現方法について説明する．

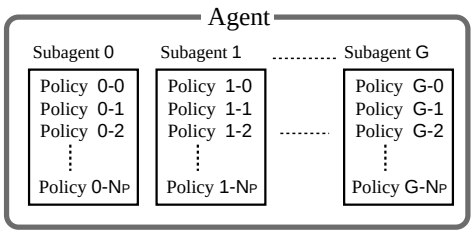


図 3 エージェントとサブエージェント
Fig. 3 Agent and subagent.

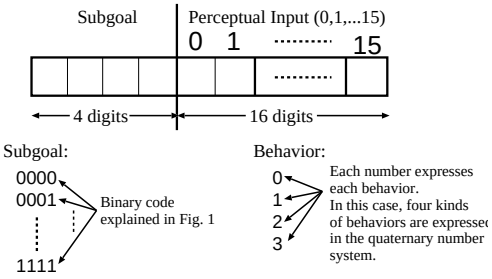


図 4 DRGA における遺伝子表現（図 1 の問題の場合）
Fig. 4 Genotype in DRGA (case of Fig.1).

4.1 DRGA － 構成

DRGA ではエージェントが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，一定の試行数が終了するとエージェント自体が遺伝的アルゴリズム（Genetic Algorithm, GA）[6] によって進化する．

DRGA は複数のサブエージェントから成り立っており，各サブエージェントは，それが適用されるべきスタート地点(注 2)の知覚情報（スタート情報）を持つ．サブエージェントは知覚状態の種類数だけ存在し，各サブエージェントは複数の政策を持つ（図 3）．政策の数はサブエージェントに共通で N_P とする．ここで，政策とは各知覚状態から行動への固定されたマッピング（$|O| \times 1$）と，サブゴールとなる知覚状態の二つをあわせたものである(注 3)．

エージェントは，サブエージェントの切替え地点において，その知覚情報に対応するサブエージェントの持つ政策の中から一つを選び，その政策に基づいてサブゴールに到達するまで行動する．サブゴールに到達したら，その知覚情報をスタート情報とするサブエージェントに切り替わる．

サブエージェントが持つ政策のレパートリは GA によって進化し，サブエージェント内での政策の選択には Q(λ)-learning [7] を用いる．GA と Q(λ)-learning は，環境中でゴールした時などに得られる報酬を時間にさかのぼって分配したもの（遅れ報酬）に基づいて行われる．サブゴール学習とサブエージェントの行動の学習は，どちらも GA によって探索範囲を絞られた上で，Q(λ)-learning によって実現されている．

4.2 政策の進化 － GA

4.2.1 コード化（encoding/decoding）

DRGA ではエージェントの知覚情報に対する決定的な行動政策が GA によって形成される．これらの政策は一種の先天的行動とみなせる．

DRGA では図 4 に示すように，一つの政策を 1 染色体として表現する．すなわち，染色体にはサブゴール情報と各知覚状態から行動へのマッピングが記述されている．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 $|O|$ 個であれば，染色体は $|A|^{|O|} \times |O|$ 種類の組合せを持つことになる(注 4)．

4.2.2 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）はサブエージェント内で行う．Goldberg [6] によると GA は building-block 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，全てのサブエージェントを一緒にして 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対応できなくなる．従って異なるサブエージェント間での GA 処理は行わない．

まず学習開始時に，各サブエージェント内でランダムに政策の初期集団（大きさ N_P）を作る．この時，明らかに壁方向に向かう政策は初期値にならないように省いている．そして，エージェントが環境中で活

（注 2）：3 章の「サブエージェントの切替え地点」と同じ．
（注 3）：4.5 節の実施例で示すように，サブゴールに到達できること自体が政策の有効性に大きく関係しているので，サブゴールを政策に含める．
（注 4）：図 4 の問題の場合は，GA の探索空間は 6.9×10^{10} の大きさを持つ．GA は非常に大きな探索空間中で準最適な解を発見するのに向いている手法なので，このような探索空間の大きい問題に GA を用いることにした．

動し，4.3 節で説明する Q(λ)-learning によって，各政策の適応度を求める．DRGA では，各政策の適応度として，政策選択の学習において獲得される Q 値（式 (5) 遅れ報酬に基づく評価値）を用いる．

GA 処理として行うのは以下のもので，全てのサブエージェントで同様に行う．

- 選択 …… エリート+ルーレット戦略
- 交叉 …… 1〜3 点交叉
- 突然変異 … 遺伝子の置換（ランダム）

まず初めに現在の政策集団から次世代の親となる N_P 本を選択する．その際，エリート戦略で適応度の高い順に N_e本を選び，残りは適応度に比例した確率で選択する．その再生された親染色体集団からランダムに２本ずつ染色体を抜き出し，C_pの確率で交叉させる．交叉とは，ランダムに交叉点（1〜3 点）を決め，交叉点の間で交互に親の遺伝コードを入れ換える処理である．その後，各染色体の各遺伝子座に対して，M_p の確率で突然変異を起こす．突然変異の際にはランダムな遺伝子に入れ換えるが，明らかに壁方向に向かう変異は起こさないように省いている．再生された親染色体集団に対してこの処理を $N_P/2$ 回繰り返すことによって，新たな世代の政策集団を形成する．

4.3 政策選択の学習 − Q(λ)-learning

各サブエージェントは，遺伝的に保持している政策のレパートリの中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(λ)-learning を使用する．

4.3.1 政策選択

DRGA では，各サブエージェントにおいて政策学習に強化学習を用いる．一般的な強化学習は行動学習手法であるが，DRGA では政策の学習に用いる．DRGA で使用する強化学習アルゴリズムは Q(λ)-learning である．DRGA での Q 値は各政策の評価値であり，**政策テーブル**で表現する．通常の Q テーブルでの状態が政策テーブルではサブエージェントに相当し，Q テーブルでの各状態における行動が，政策テーブルでは各サブエージェントにおける政策に相当する．DRGA では，知覚状態から行動へのマッピングが決定的であるため，あるサブエージェントで政策を選んだ時にサブゴールに到達するかどうかは環境に対して決定的である．

強化学習を行う場合，通常，常に最大 Q 値の行動を選択するのではなく，ある程度の探索機能を残しながら行動を選択する．選択方策としては確率 $prob$ で最大の評価値を持った行動を選び，確率 $1-prob$ 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 [8]．DRGA では次の式 (3) で示されるボルツマン分布による選択を用いる．

$$prob_g(p) = \frac{e^{Q(g,p)/T}}{\sum_{p' \in P_g} e^{Q(g,p')/T}} \tag{3}$$

$prob_g(p)$ はサブエージェント g でサブエージェント内の政策 p を選ぶ確率である．P_gはサブエージェント g 内の政策集合を表す．$Q(g,p)$ はサブエージェント g における政策 p の評価値，すなわち Q 値である．T は政策を選ぶ際のランダム性を調整する温度パラメータである．

4.3.2 学習更新式

Q 値の更新には offline Q(λ)-learning を用いる．時刻 i においてサブエージェント g_iで政策 p_iを実行した時にシステムが Q 値を更新する式を以下の式 (4)，式 (5) に記す．

$$Q'(g_i,p_i) \leftarrow R_i + \gamma[(1-\lambda) \max_{p' \in P_{g_{i+1}}} Q(g_{i+1},p') + \lambda Q'(g_{i+1},p_{i+1})] \tag{4}$$

$$Q(g_i,p_i) \leftarrow (1-\alpha)Q(g_i,p_i) + \alpha Q'(g_i,p_i) \tag{5}$$

ここで $i = 1,2,...$ 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_iはサブエージェント g_iの得た報酬の総計である．γ は割引率，α は学習率で，共に $0 \leqq \gamma \leqq 1$，$0 \leqq \alpha \leqq 1$ である．$Q'(g_i,p_i)$ は eligibility traces（2.2 節注１参照）を用いた場合の目指すべき Q 値であり，λ ($0 \leqq \lambda \leqq 1$) は eligibility traces を用いる程度を表す定数である．

学習の更新は $i = I, I-1, .., 2, 1$ の順に計算する．I 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．エージェントが終了条件を満たすまでを１試行と呼ぶ．式 (4) を計算し式 (5) に代入し，$Q(g_i,p_i)$ を更新する．この時，時刻 I では $Q'(g_I,p_I) = R_I$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（$MaxStep$）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て，その時の知覚情報に応じて次のサ

ブエージェントに切り替わる．タスクを達成すると大きな報酬を得る．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時もあれば，複数の政策で寿命を使い切る時もある．

4.4 DRGA − 全体像

図5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（$SIZE_T$）を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における個体表現はDRGAでのサブエージェント内の政策にあたるため，進化の評価基準である適応度を各政策のQ値で表現し，GAを用いて進化を行う．進化後，政策テーブルの値を初期化して，また新たな政策集合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．

各政策はサブゴールに到達した時に微小な報酬を得るので，強化学習によって各サブエージェント内ではサブゴールに到達できる政策の選択確率が高まる．サブゴールに到達できる政策は，環境中である程度の距離を移動していることが多いので，一度もゴールに辿り着いていない段階での探索範囲の拡大に役立つ．また一度ゴールに到達すると大きな報酬が得られるので，ゴール到達に役立つ政策は強く強化される．この強化学習の評価値 $Q(g,p)$ を適応度としてGAを行うので，サブエージェント中では，タスク全体を見て色々な場面で役に立つ政策が次第に優位になってくる．学習初期の段階では，サブエージェント内で政策の競合が起こる場合もあるが，GAの近傍探索能力で行動やサブゴールを少しずつ変化させ，環境中では競合の起こらない経路を見つけ，強化していくことによって，次第に競合を解消する．そして，各サブエージェントに一つの優位な政策が生き残り学習が収束する．

4.5 DRGA 実施例

DRGAの動きを図6の10×10迷路を使って説明する．タスクは初期位置Sから目的地Gに到達することであり，そのタスクを達成する経路を獲得することがDRGAの目的である．エージェントが行えるのは「北へ1マス移動，東へ1マス移動，南へ1マス移動，西へ1マス移動」の4行動である．エージェントが得られる知覚情報は隣接したグリッドに壁があるかないかだけである（図1参照）．この限定された知覚情報では得られる知覚種類が16種類しかなく，知覚的

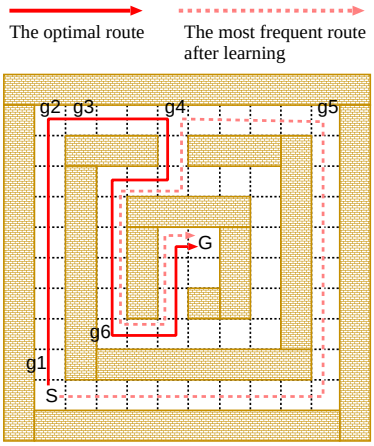


図6　10×10迷路
Fig. 6　10×10 maze.

見せかけが発生している．知覚的見せかけによりこの問題はMDPではなくなっていて，1サブエージェントでは解くことができない．例えば，図中に実線で示された最短経路をとる場合，東西に壁がある知覚状態 '0101' では，最初に北へ移動（S→g4），次に南へ移動（g4→g6），最後に北へ移動（g6→G）しなければGには辿り着かない．この問題では，最小で二つのサブゴールを置かなくてはならない．

この問題をDRGAで解く場合，まずSでは南西に壁がある知覚情報 '0011' をスタート情報とするサブエージェントが起動する．このサブエージェントの政策集団から式(3)によって一つの政策が選択される．学習開始時には全ての政策のQ値が0なので，全くランダムに選ばれることになる．選ばれた政策のスタート情報における行動が北進だった場合は実線の方向に一マス（g1の位置），東進だった場合は破線の方向に一マス進む．政策集団を初期化する際に明らかに壁に進む政策は除いているので，ここで南進や西進する政策が選ばれることはない．

北進した場合の続きを考える．次に得られる知覚情報は '0101' で，これに対する現在の政策の行動は北進と南進とこの知覚情報をサブゴールとする場合の3通りがあり得る．南進だった場合は再びSに戻る．政策の行動は決定的なので，この政策は「北南北南…」という行動を $MaxStep$ まで繰り返して報酬を受けとらないまま寿命が尽きる．一方，北進だった場合には，北に一マス進み，また同じ知覚情報なので北に一マス進むというのを繰り返して，g2の位置に到達す

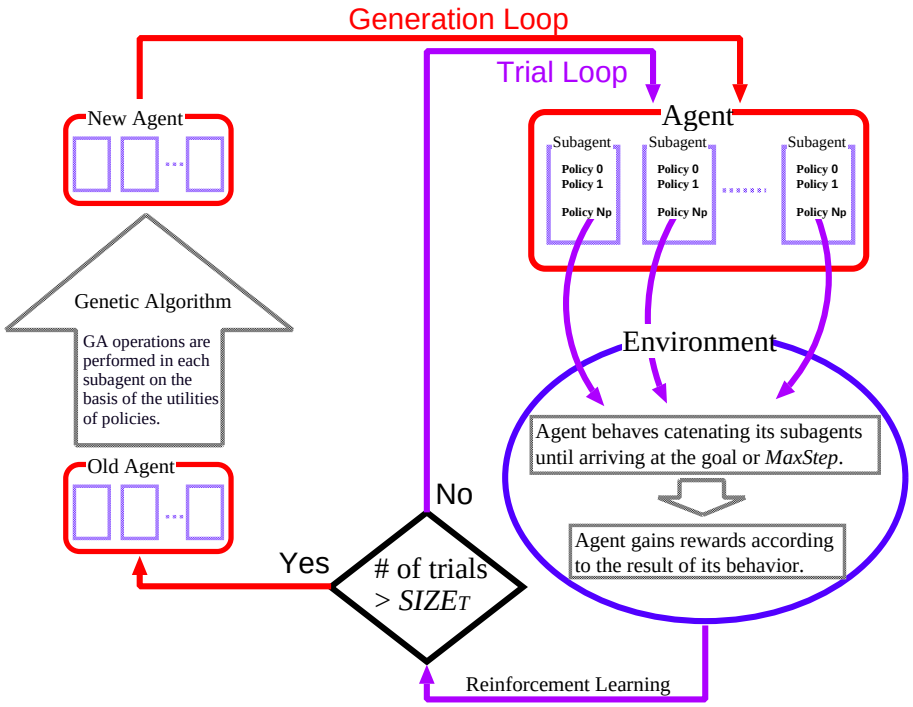


図 5　DRGA 全体像
Fig. 5　Outline of DRGA.

る．ここで得られる知覚情報は ‘1001’ で，これに対する行動は南進と東進とこの知覚情報をサブゴールとする場合の 3 通りがあり得る．南進だった場合には一マス南に戻り，そこでの行動は必ず北進なので，「南北南北…」という行動を $MaxStep$ まで繰り返して報酬を受けとらないまま寿命が尽きる．東進だった場合にはさらに実線の経路を辿っていくことになる．

このように考えると，Sから北進する政策のうちサブゴールまで到達するのは，表 1 に示す 5 種類である．表にない 10 種の知覚情報に対しては任意の行動で良い．p1〜p4 はそれぞれ実線上を通って g1〜g4 まで，p5 は g4 を通過して g5 まで到達して微小な報酬を受けとり，次のサブエージェントに制御を譲る．このようにサブゴールまで到達できる政策は限られていて，微小な報酬によって次第に優勢になってくる．

一つ目のサブゴール以降のサブエージェントやSから東進した場合も，同様にサブゴールまで到達できる政策が次第に優勢となる．そしてそのような政策の組合せでGまで到達した時，大きな報酬を受けとり，Q(λ)-learning の報酬の分配によって経路上の全ての政策が強く強化される．Q 値に基づく GA による淘汰によって，有効な政策がますます優勢となる．

表 1　サブゴールまで到達する政策
Table 1　Policies that can reach a subgoal

政策	南西壁 0011	東西壁 0101	北西壁 1001	南北壁 1010	北　壁 1000	北東壁 1100
p1	北進	sg	＊	＊	＊	＊
p2	北進	北進	sg	＊	＊	＊
p3	北進	北進	東進	sg	＊	＊
p4	北進	北進	東進	東進	sg	＊
p5	北進	北進	東進	東進	東進	sg

sg: サブゴール
＊: 任意の行動

ここで，ゴール到達時の大きな報酬として次の式 (6)，サブゴール到達時の微小な報酬として次の式 (7) を用いることを考える．

$$R_i = (MaxStep - step)^2/10.0 \qquad (6)$$

$$R_i = (observenumber) \qquad (7)$$

$step$ は初期位置からゴール到達までのステップ数，$observenumber$ はサブエージェントがサブゴールに到達するまに知覚した知覚情報の種類数である．同じ知覚状態の観測を続けても $observenumber$ は増えず，新しい知覚情報を得た時に，その数をカウント

し，これをサブゴール到達時の微小報酬とする．

表 1 の 5 種の政策の場合，サブゴール到達時の報酬は下にいくほど大きくなり，pn に対して $n+1$ の報酬となる．一方，ゴール到達時の報酬はそれぞれのサブゴールから最短経路を通ったと仮定すると，p1〜p4 を使用した場合は 518.4，p5 を使用した場合は 384.4 となる．こうして，ゴールまでの最短経路上にあり，かつ長い距離を移動している p4 が最も強化されることになる．

ただし，この最短経路はサブゴールでの知覚的見せかけによってゴール到達が困難になる経路となっている．それについては次節で詳述する．

4.6 DRGA の効果と欠点

前節の 10×10 迷路のタスクを例として，DRGA の本質的な効果と欠点を説明する．DRGA では，タスク達成にとって障害となる知覚的見せかけを排除し，障害とならない場合は逆にうまく利用して，大規模問題を複数の類似した小問題に分ける．そのため，必ずしも最適解に収束するとは限らない．

10×10 迷路のタスクでは，図 6 の g4 と g6 の地点にサブゴールを置いた実線の経路が最適解（最も報酬の高い経路）である．Sと g6 は同じ知覚情報 '0011' を持ち，同じサブエージェントが起動する．しかし，Sでは北進，g6 では東進しなければならないため，このサブエージェント内では政策の競合が起こり，うまくゴールに到達できないことが多い．それに対し，g5 と g6 の地点にサブゴールを置いた破線の経路の場合は，Sから g5 までと g6 からGまでは同じ政策を再利用することが可能で，政策の競合も起こらない．この場合，ゴールに到達するために二つの政策だけを学習すれば良いので，学習効率も良い．実際，このタスクを DRGA で学習してみたところ，学習収束時には 74.8%の高い確率でこの経路が選択された．

この例に示すように，DRGA では有効な政策の再利用によって，ゴールまでに必要な政策の数を減らす機能がある．そのため，ゴールまでに多数のサブエージェント切替えを必要とするタスクに対しても，再利用可能な有効な政策を見つけることによって探索空間を制限し，うまく学習することが期待できる．

一方，欠点としては，タスク達成にとって障害となる知覚的見せかけを排除するために，必ずしも最適解に収束しないことが挙げられる．知覚的見せかけが障害となる場合，サブエージェント内で政策の競合が起こっている．その場合には，学習の進行とともに，GA の近傍探索能力で行動やサブゴールを少しずつ変化させ，環境中では競合の起こらない経路を見つけ，強化していくことによって，次第に競合を解消する．そして，各サブエージェントに一つの優位な政策が生き残り，学習が収束する．このように知覚的見せかけがタスク達成の障害となる場合でも，準最適解を見つけて知覚の見せかけ問題を回避する機能を持っている．

また，障害となる知覚的見せかけを排除することができない場合もあり得る．ゴールまでの経路が限られていて，競合の起こらない経路が存在しない場合などである．例えば図 6 の迷路において，g5 の位置が壁に塞がれている場合がそうである．そのような問題は探索問題としてもともと難しい問題で，スタート地点の情報だけではなく，その前後の状態変化に基づいてサブエージェントを選ぶような手法が考えられるが，そのような柔軟な政策の切替えは今後の課題である．

HQ-learning と比較してみると，HQ-learning におけるサブゴール学習は，問題を MDP に分割することのみが目的であるが，DRGA におけるサブゴール学習は，それに加えて，有効な政策がうまく連鎖することも目的としている．そのために DRGA のサブゴール学習は少し制約が大きい．一方，サブゴール間の政策については，DRGA では再利用が可能なので，各々独立に学習する HQ-learning と比較すると学習が非常に効率的である．また，DRGA では，ゴールまでに必要なサブエージェントの数をあらかじめ予想する必要がないという点も利点である．

5. シミュレーション実験

5.1 タスク (a) − Key and door タスク

5.1.1 タスク設定

一つ目の実験として，図 7 に示す 26×23 迷路のタスクを扱う．このタスクは，HQ-learning の元論文 [2] において挙げられていた最も複雑な問題である．この実験では，このタスクを用いて，HQ-learning で解ける程度の複雑な問題を DRGA でも解くことができることを示す．

このタスクは，初期位置Sから始まり，(1) 位置Kにある鍵をとって，(2) 鍵を使ってドア（グレー部分）をくぐり抜けて，(3) Gへ到達することである．この迷路においてエージェントが得られる知覚情報は 11 種類である．この迷路は鍵を取らないとドアをくぐり抜けられない．そこで，鍵を持っていることを区別す

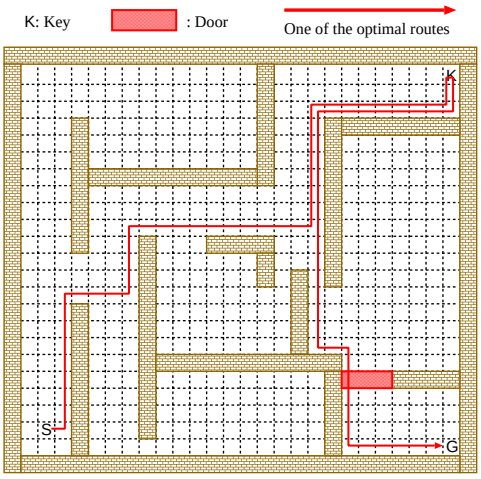


図 7　Key and door タスク
Fig. 7　Key and door task.

ると，全状態数は 960 状態存在する．また，Gまでの最短経路は 83 ステップ，必要最低サブゴール数は 2 である．従って，政策の再利用の可能性はほとんどない(注 5)ので，再利用性の効果の検証は行わない．

文献 [2] によると，1000 ステップを限界とするランダムウォークでは，20000 試行行って 1 回もゴールに到達することが無かった．また，10000 ステップを限界とするランダムウォークでは，3000 試行行って最短経路は 1174 ステップだった．

5.1.2　報酬およびパラメータ設定

報酬は 4.5 節で説明したものを用いる．すなわち，ゴール到達時には式 (6)，サブゴール到達時には式 (7) の報酬を与える．

各パラメータは，GA に関しては $N_P = 30, N_e = 4, C_p = 0.9, M_p = 0.15$ と定める．交叉は 2 点交叉のみとする．Q(λ)-learning に関しては $\gamma = 0.7, \alpha = 0.15, \lambda = 0.4$ とし，各政策の Q 値の初期値は 0.0 と定める．行動選択のための温度パラメータ T は 3.0 とする．全体に関しては $MaxStep = 250$ とし，世代更新するための試行回数 $SIZE_T$は，初期値を 1000 とし，世代ごとに 100 ずつ，7500 まで増加させる．この時の世代は 65 世代目であり，その後の世代では 7500 のままとする．実験終了までの世代数（generation）は 100 とする．

(注 5)：スタート情報とサブゴールが等しい政策はスタートできず役に立たないので，サブゴールの前後の政策は必ず異なる．従って，サブゴールが二つしかない場合は，一つ目の政策と三つ目の政策が等しくなる組合せ以外に政策の再利用の可能性はない．

表 2　各ステップ数の割合 — Key and door タスク
Table 2　Ratio of each step - Key and door task.

-	DRGA	HQ-learning
No.1	47.0%(87step)	40.0%(87step)
No.2	3.63%(89step)	21.0%(89step)
No.3	2.95%(85step)	8.00%(91step)
タスク失敗	31.1%(250step)	8.00%(1000step)

あわせて，文献 [2] より HQ-learning のパラメータも載せておく．ゴール到達時の報酬を 500，ゴールしない全ての行動に -0.1 の報酬を与える．Q 値更新には DRGA の式 (4)，(5) と同様の Q(λ)-learning の更新式，HQ 値更新には式 (2) を用いる．$\gamma = 1.0, \lambda = 0.9, \alpha = 0.05, \alpha_{HQ} = 0.1, T = 0.2, MaxStep = 1000$ と定める．P_{max}は最初の試行では 0.4 で，最後の試行までに線形的に 0.8 に上昇させる．γ, λ, P_{max}は Q 値の式と HQ 値の式で共通である．ここで，$MaxStep$（エージェントの寿命）が DRGA と大きく異なるが，DRGA は行動にランダム性が全くなく，政策選択時にのみランダム性がある．このため，寿命である $MaxStep$ を多く取る必要性がない．

5.1.3　結果 — タスク (a)

図 8 に経路長分布図を示す．HQ-learning のグラフは 8 エージェントシステムで行った実験の結果である．HQ-learning では 60000 試行行っていて，600 試行毎のゴールまでのステップ数の分布を示している．この実験は 100 回の実験の平均である．また，実験が終了した時点（DRGA は 100 世代目，HQ-learning は最後の 600 試行）での，経路長分布図の最も割合を占めたステップ数から上位三つとタスクを失敗した割合を表 2 にまとめる．図 8 と表 2 より DRGA，HQ-learning ともに POMDP を解決していると言える．

5.1.4　考察 — タスク (a)

DRGA，HQ-learning 共に過半数以上がゴールに到達している（学習終了時にゴールに到達出来なかった割合が DRGA は 31.1%，HQ-learning は 8.00%）ので，共にこのタスクの難易度の POMDP を解決していると言って良い．DRGA では最終的なタスク失敗率がかなり大きいが，これは世代更新時に政策テーブルがクリアされ，新しく始まった世代の初期では，ランダムに政策が選ばれるので試行の失敗率が高くなることが一つの原因となっている．しかし，100 回の実験においてタスクを成功させるサブエージェントが形成されなかったケースは一度もなく，DRGA で知

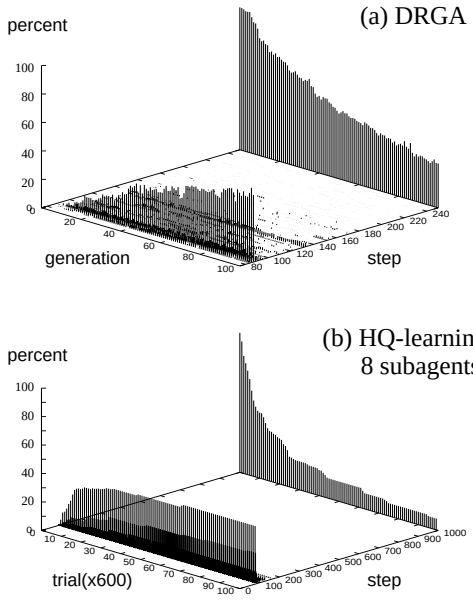


図 8　経路長分布図 － Key and door タスク
Fig. 8　Histogram of the number of steps to the goal.

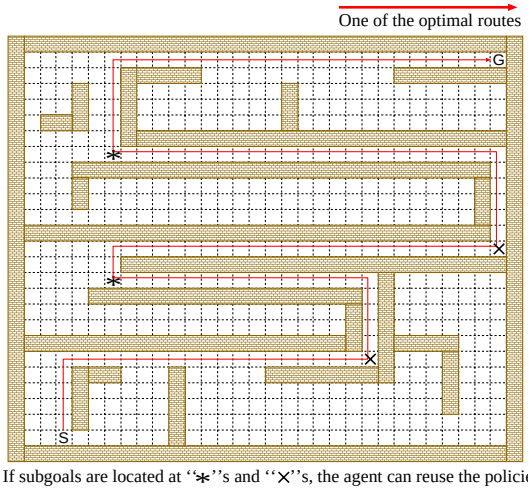


図 9　30×25 迷路
Fig. 9　30×25 maze.

覚の見せかけ問題を解決できているといえる．

このタスクでは，1000 ステップまでのランダムウォークでは 20000 試行で一度もゴールに到達しなかったのに，DRGA では 250 ステップ，HQ-learning でも 1000 ステップの試行を繰り返すことで適切な政策が獲得できている．これは，行動のランダム性を制限して，直進性を高めているためだと考えられる．

このようにゴールまでに多くのステップ数を必要とするタスクでは，学習の開始時点において全くのランダムウォークでは同じ場所を繰り返し探索してなかなか探索範囲を広げられないし，万が一ゴールに到達しても，ゴールまでのステップ数が多すぎてうまく報酬を分配することができない．エージェントに直進性を持たせると，探索範囲が広がりやすく，探索効率が良くなると期待できる．

DRGA は各知覚状態に対する行動を固定しているので直進性が非常に高い手法であり，このようなゴールまでの経路が長いタスクを解く場合に有効である．HQ-learning も，2.2 節で述べたようにタスク終了時に一括して Q 値の更新を行うことによって，行動の直進性を高めている．すなわち，このようなオフライン更新では，試行中に Q 値に変化がないので，同じ知覚状態において最も Q 値の大きい行動を繰り返し選択する可能性が高いのである．このタスクにおいては，全ての行動に微小な負の報酬を与えることによって，さらに直進性を高める工夫をしている．すなわち，この負の報酬によって，一度もゴールに到達していない段階でも Q 値の差が出てきて，Q 値の大きい行動を繰り返す可能性が高まるのである．

5.2　タスク (b) － オリジナル迷路（30×25 迷路）

5.2.1　タスク設定

二つ目の実験として，先程の Key and door タスクよりも難易度の高いオリジナル迷路を作成した（図 9）．このタスクを用いて，DRGA の有効性，すなわち有効な政策の再利用によって，大規模な部分観測マルコフ決定問題が解けることを示す．

この迷路は 30×25 の大きさでタスク (a) の 26×23 の迷路よりも広いが，鍵，ドアなどが無いため状態空間としては小さくなる．タスク (a) が状態数 960 だったのに対して，この迷路は状態数 551 である．しかし，DRGA や HQ-learning にとっての難易度とはゴールまでに必要なサブエージェントの数と経路の長さに主に依存している（5.2.3 節参照）．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替えを最低でも 4 回行わなくてはならない．最短経路は 131 ステップである．従って，タスク (a) よりもかなり難易度が高い．

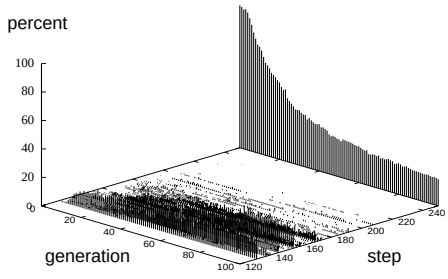


図 10　経路長分布図 － 30×25 迷路（DRGA）
Fig. 10　Histogram of the number of steps to the goal.

表 3　各ステップ数の割合 － 30×25 迷路（DRGA）
Table 3　Ratio of each step - DRGA.

-	DRGA
No.1	24.0%(145step)
No.2	18.2%(131step)
No.3	9.73%(155step)
タスク失敗	18.8%(250step)

5.2.2　結果 - タスク (b)

タスク (a) と同じ報酬やパラメータを用いて，DRGA にこのタスクを学習させた．図 10 に経路長分布図を示す．この図もタスク (a) の図 8 と同様に 100 回の実験の平均である．また，表 3 に実験が終了した時点（100 世代目）での，経路長分布図の最も割合を占めたステップ数から上位三つとタスクを失敗した割合を示す．図 10 と表 3 により，DRGA は十分にこのタスクの POMDP を解決していると言える．100 世代目のエージェントではタスクを達成できない試行が 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．

5.2.3　考察 － タスク (b)

DRGA のように，不完全知覚の迷路問題を MDP に分割して解く手法では，学習空間は $(|A|^{|O|} \times |O|)^M$ の大きさを持つ．ただし，$|A|$ は行動の種類数，$|O|$ は知覚情報の種類数，M はゴールまでに必要なサブエージェントの数である．$|A|, |O|$ は定数なので，このうち重要なのは M である．また，タスクの難易度には，ゴールまでの最短経路の長さ（l_{min}）も大きく関係している．ゴールまでの経路が長いと，当然，必要なサブエージェントの数も増える傾向があり，さらに，(1) 最初に偶然にゴールに到達するのが非常に困難になる点，(2) サブゴール間の経路も長くなって，サブエージェントの冗長性が低くなり（表 1 の＊のように任意の行動で良い知覚状態が少なくなる），行動の学習が困難になる点もタスクの難易度を増す．

タスク (b) は $M = 5, l_{min} = 131$ とどちらも大きく，タスク (a)（$M = 3, l_{min} = 83$）と比較すると，かなり難易度が高い．しかし DRGA では，学習後のこのタスクの失敗率は 18.8%と，タスク (a) と比較しても小さくなっている．これは，スタート情報に基づくサブエージェント選択によって政策の再利用を行っているためである．DRGA では，図 9 の×と＊にサブゴールを置くことによって，×でスタートする政策と＊（および S）でスタートする政策の二つの政策のみでゴールすることができる．従って，$(4^{16} \times 16)^5$ の探索空間を，実質 $(4^{16} \times 16)^2$ まで限定して解くことができる．タスク (a) の探索空間は $(4^{16} \times 16)^3$ であり，この差がタスク失敗率の逆転現象につながっている．少し恣意的な迷路構造ともいえるが，迷路構造に関して何のアプリオリな知識も与えておらず，$(4^{16} \times 16)^2$ の探索空間は環境との試行錯誤した結果うまくサブゴールを配置することで得られたものである．DRGA では環境に応じてサブエージェントを決定する適応性によって，このような大規模な問題を簡単にして扱う機能がある．

l_{min} が大きくなったことによる (1),(2) の問題点は避けられないが，(1) に関しては，DRGA の行動の高い直進性によって，このタスクでも 250 ステップの行動の繰り返しでうまくゴールを発見することができている．

5.2.4　HQ-learning との比較

比較のために，HQ-learning でもこの 30×25 のオリジナル迷路のタスクを学習させてみた．報酬とパラメータに関してはタスク (a) と同じとしたが，サブエージェント数は 16 で設定した．このタスクでは，ゴールまでに最低 5 個のサブエージェントが必要であり，16 という数は多少余裕を持たせるためである．HQ-learning においてサブエージェント数を増やすと，タスクに対する適応力は高くなるが，サブエージェントの数だけ Q テーブルが存在するので，学習の収束時間が増加する．DRGA ではサブエージェントの切替え数をあらかじめ定める必要はないので，このような適応性と収束時間のトレードオフを考慮する必要はない．

図 11 に経路長分布図を示す．HQ-learning では 120000 試行行っていて，600 試行毎のゴールまでの

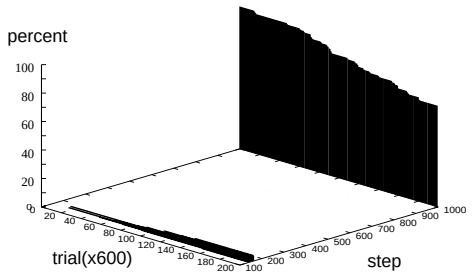


図 11　経路長分布図 － 30×25 迷路（HQ-learning）
Fig. 11　Histogram of the number of steps to the goal.

表 4　各ステップ数の割合 － 30×25 迷路（HQ-learning）
Table 4　Ratio of each step - HQ-learning.

-	HQ-learning
No.1	4.00%(161step)
No.2	4.00%(164step)
No.3	3.00%(153step)
タスク失敗	71.0%(1000step)

ステップ数の分布を示している．この図もタスク (a) の図 8 と同様に 100 回の実験の平均である．また，表 4 に最後の 600 試行の，経路長分布図の最も割合を占めたステップ数から上位三つとタスクを失敗した割合を示す．

図 11 と表 4 により，HQ-learning はこのタスクの POMDP を解決しているとは言い難い．120000 試行の段階で行動のステップ数としては合計 10661 万ステップであり，DRGA の 100 世代までの合計 9701 万ステップと比較しても，十分に長時間学習しているといえる．図 11 で，まだタスク失敗率が線形的に減少しているように見えるのは，行動のランダム性を線形的に減少させていることが主な原因であると考えられる．実は，この実験は初め，タスク (a) と同じ 60000 試行で行ったのだが，この時も同様にタスク失敗率は線形的に減少し，最終的には 67.0% だった．試行数を倍に増やして再実験を行った結果が 71.0%と改善が見られないので，これ以上試行数を増やしても，大きな性能の向上は期待できない．

HQ-learning では政策の再利用を行っていないので，タスク (b) の探索空間は $(4^{16} \times 16)^5$のままであり，タスク (a)（$(4^{16} \times 16)^3$）と比較するとかなり大きい．HQ-learning でタスク (a) が解けたのに対してタスク (b) が解けなかったのは，難易度通りの結果といえる．それに対し提案手法である DRGA では，うまく政策の再利用を行って，探索空間を限定することでタスク (b) のような大規模な問題を解決することができた．

6. む す び

本論文では，部分観測マルコフ決定問題において色々な場面で同一の小問題がたくさん現れるような問題を対象として，DRGA を提案した．DRGA は，遅れ報酬に基づく GA と強化学習の組合せで，サブゴールの配置の学習とサブゴール間の行動政策の学習を同時に行う．DRGA では，タスク全体を見て問題を MDP に分割しながら，色々な場面で役に立つ政策が生き残り，再利用可能な場面を見つけてそれらの有効な政策を適用していくことによって大規模な部分観測マルコフ決定問題に対応する．シミュレーション実験を通して，従来の同様な POMDP をサブタスクに分割して解く手法では解けなかった大規模な問題を DRGA が解けることを示した．

DRGA では，政策を適用するかどうかの判断を，スタート地点の知覚情報に基づいて行っている．そのため，スタート地点の知覚的見せかけによって解けないような問題も存在する．スタート地点の知覚的見せかけが回避できないことが判断できた場合には，その前後の状態変化など別の指標に切替えて知覚的見せかけを回避するのが望ましい．そのような柔軟な政策の切替えは今後の課題である．また，DRGA で扱える POMDP は，分割すれば MDP となるものだけである．今後，実問題として，そのような性質を持つものに本手法を適用することを目指している．

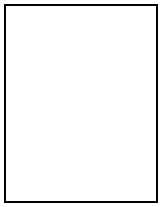
謝辞　なお，本研究の一部は文部省科学研究費基盤研究（B）（2）課題番号 11480078 の援助によって行われた．

文　　献

[1] S. D. Whitehead and D. H. Ballard, "Learning to perceive and act by trial and error," Machine Learning, vol.7, pp.45–83, 1991.
[2] M. Wiering and J. Schmidhuber, "HQ-learning," Adaptive Behavior, vol.6, no.2, pp.219–246, 1997.
[3] C. J. C. H. Watkins and P. Dayan, "Technical note: Q-learning," Machine Learning, vol.8, pp.279–292, 1992.
[4] L.-J. Lin, Reinforcement Learning for Robots Using Neural Networks, PhD thesis, Carnegie Mellon University, Pittsburgh, 1993.
[5] R. S. Sutton and A. G. Barto, Reinforcement Learning: An Introduction, The MIT Press, Cambridge, 1998.

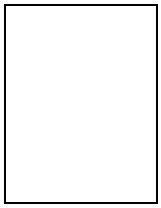
[6] D. E. Goldberg, Genetic Algorithms in Search, Optimization, and Machine Learning, Addison-Wesley, 1989.
[7] J. Peng and R. Williams, "Incremental multi-step Q-learning," Machine Learning, vol.22, pp.283–290, 1996.
[8] S. D. Whitehead and D. H. Ballard, "Active perception and reinforcement learning," Proc. of 7th International Conference on Machine Learning, pp.162–169, 1990.

（平成年月日受付，月日再受付）



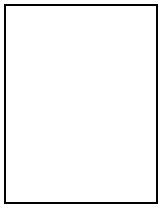
山城　啓秀

平 10 大阪教育大学・教・情報卒，平 12 奈良先端科学技術大学院大学・情報科学研究科修士課程了．同年三洋電機入社．在学中は人工知能，とりわけ強化学習，遺伝的アルゴリズムの研究に従事．



上野　敦志

平 3 東京大学工学部航空学科卒業．平 8 同大学院航空宇宙工学専攻博士課程単位取得退学．同年，奈良先端科学技術大学院大学情報科学研究科助手，現在に至る．博士（工学）．ロボットの学習，自律システムの研究に従事．人工知能学会会員



武田　英明　（正員）

昭 61 東京大学・工卒，平 3 東京大学大学院工学系研究科博士課程修了．財団法人日本システム開発研究所嘱託研究員，ノルウエー工科大学 Postdoctoral Fellow，奈良先端科学技術大学院大学院助手，助教授を経て現在国立情報学研究所知能システム研究系助教授．工学博士．知的 CAD，人工知能，設計学の研究に従事．人工知能学会，電子情報通信学会，日本ロボット学会，精密工学会，AAAI 各会員．人工知能学会誌編集委員．電子情報通信学会論文誌編集委員．