1D3-4

# POMDP 環境下での強化学習における GA による サブゴールの動的生成

Dynamic Subgoal Generation Using GA for Reinforcement Learning under POMDP

野村 拓己[*1] 加藤 昇平[*2]
Takumi Nomura Shohei Kato

[*1]名古屋工業大学 大学院工学研究科 情報工学専攻
Dept. of Computer Science and Engineering, Graduate School of Engineering, Nagoya Institute of Technology

In this paper, we will propose a method generating sub-goal for reinforcement learning for POMDP. POMDP is an environment where an agent gets confused by several states even when same information is observed from the environment. To resolve this problem we will propose a genetic algorithm that dynamically generates sub-goal for reinforcement learning. the number of sub-goals are not tuned for our method, and each of the agents has different solutions since they behave independently.We confirmed the effectiveness of our method by some experiments with partially observable mazes with HQ-Learning.

## 1. はじめに

近年，強化学習の研究が盛んに行われている．強化学習は，学習エージェントが試行錯誤を通じて制御則を獲得する機械学習の一種である．学習エージェント自身が制御則を学習・獲得するため，効率的な制御則を発見する可能性も考えられる．そのため，ロボットの自律的な行動獲得などにおいて強化学習を用いた研究が盛んに行われている．強化学習では知覚した観測情報から環境を一意に特定して学習が行われる．そのため，異なる環境から知覚した観測情報が同一である場合，それらを異なる環境を同一の環境と認識し学習する [1]．同一と認識された個々の環境において最適な行動が異なる場合，学習が混同してしまい，正しく学習が行えない．このような問題を持つ環境を部分観測マルコフ決定過程（POMDP）と呼ぶ．

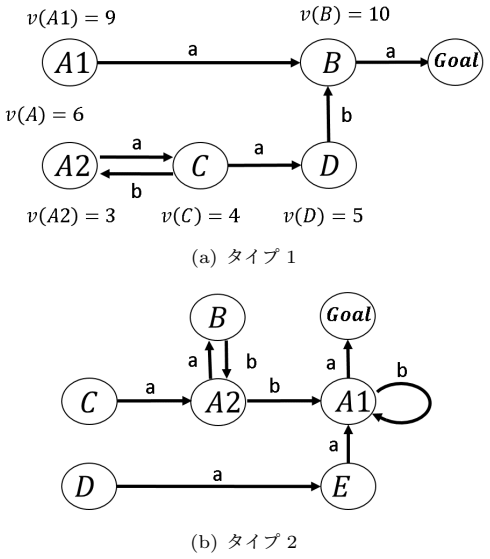
POMDP 環境下における問題解決手法として HQ-Learning が提案されている [2]．HQ-Learning は学習すべきタスクをサブタスクに分割し，それぞれにサブゴールを設定し，別環境にもかかわらず観測情報が同一となる状況を回避している．HQ-Learning ではサブゴールを動的に探索している．しかし HQ-Learning は多くの問題点がある．例えば，得られる解が１通りであるため環境変化に脆弱である．またサブゴール数を予めユーザが与える必要がある．さらに学習が進むにつれ新しいサブゴールの発見が難しくなる．

そこで本研究では，POMDP 環境下での新たな問題解決手法として，遺伝的アルゴリズム（GA）を用いたサブゴールの動的生成手法を提案する．提案手法において，エージェントは独立したサブゴールを持つため，複数解の発見が可能となる．またサブゴール数を可変にすることで，ユーザが予めサブゴール数を与える必要がない．さらに本研究ではサブゴールの抽象化を行い，学習初期の学習速度の向上を試みている．本稿では HQ-Learning との比較実験により，提案手法の有効性を検証する．

## 2. POMDP

POMDP では不完全知覚により 2 通りの混同が存在する [3]．例を図 1 のネットワークで示す．ノードは状態を示す．ただし状態 A1 と A2 は観測情報が共に状態 A として観測される．アークは行動の種類，$v(X)$ は状態 $X$ での状態価値を示す．

連絡先: 加藤昇平，名古屋工業大学，愛知県名古屋市昭和区御器所町，052-735-5625，shohey@juno.ics.nitech.ac.jp




(a) タイプ 1

(b) タイプ 2

図 1: POMDP における混同

### 2.1 タイプ 1

タイプ 1 の混同は価値の高い状態と価値の低い状態が同一視されるときに起こる．図 1(a) では状態 $A1$ の価値が 9，状態 $A2$ の価値が 3 である．状態 $A1$ と $A2$ は異なる状態であるが，学習器には同一の状態（状態 $A$）として観測される．したがって，状態 $A1$ と $A2$ を等確率で経験した場合，状態 $A$ の価値は平均され 6 となり，状態 $D$ の価値である 5 よりも高くなる．その結果，状態 $C$ では行動 b が最適とされ，状態 $A2$ と $C$ を往復する非合理的な政策が学習される．

### 2.2 タイプ 2

タイプ 2 の混同は合理的ルールと非合理的ルールが同一視されるときに起こる．図 1(b) において状態 $A2$ では行動 a は非合理的ルールであるが，$A1$ では行動 a は合理的ルールとな

る．学習器は $A1$ と $A2$ を同一の状態（状態 $A$）として観測するため，状態 $A$ で行動 a は学習器にとって合理的ルールとされる．その結果 $A2$ と $B$ を往復する非合理的な政策が学習される．

## 3. HQ-Learning

HQ-Learning では学習すべきタスクをマルコフ決定過程（MDP）に従うように複数のサブタスクに分割する．サブタスク内では状態が MDP に従うため従来の強化学習手法によって学習可能である．エージェントには各サブタスクを担当するサブエージェント $i$（$i = 1, \cdots, \mathrm{L}$）が存在し，各々が Q テーブルと HQ テーブルを所持する．サブタスクはサブゴール到達を目指す．観測情報は配列構造で入力され，サブゴールは観測情報と同じ長さの配列構造を持つ．

Q テーブルは観測情報に対する行動の価値，HQ テーブルはサブゴールの価値を表す．まずサブエージェント 1 が作動する．$\mathrm{HQ}_1$ テーブルをもとに Max-Random 法でサブゴールを選択する．Max-Random 法とは，$P_{max}$ の確率で $\mathrm{HQ}_1$ からサブゴールの価値が最大となるものを選択し，$(1 - P_{max})$ の確率でランダムに選択する．サブエージェントは選択されたサブゴールをゴールとみなし，Q 学習 [4] を行う．行動の選択には $\mathrm{Q}_1$ テーブルをもとに Max-Boltzmann 法を使用する．Max-Boltzmann 法とは，$P_{max}$ の確率で $\mathrm{Q}_1$ の最大のものを選択し，$(1 - P_{max})$ の確率で式（1）でボルツマン分布に基づく確率で行動を選択する．

$$prob_s(a) = \frac{exp(Q(s,a)/T)}{\sum_{a' \in A} exp(Q(s,a')/T)} \tag{1}$$

ここで，$prob_s(a)$ は状態 s で行動 a を選択する確率，$Q(s,a)$ は状態 $s$ における行動 $a$ の価値，A は取りうる行動の集合，$T$ は行動を選択する際のランダム性を調整する温度パラメータである．

サブエージェント $i$ がサブゴールに到達したとき，サブエージェント $i+1$ に制御が移される．最後のサブエージェント L がゴールに到達したとき，全てのサブエージェントの HQ テーブルと Q テーブルを更新する．それぞれ式（6）に従い一括学習を行う．一括学習とは，行動開始から状態遷移，行動選択をすべて記録しタスク終了時に一括にテーブルの更新を行う．

$$\begin{aligned} Q'(s_t,a_t) &\leftarrow r_t + \gamma[(1-\lambda) \max_{a' \in A_{t+1}} Q(s_{t+1},a') \\ &\quad +\lambda Q'(s_{t+1},a_{t+1})] && (2) \\ Q(s_t,a_t) &\leftarrow (1-\alpha)Q(s_t,a_t) + \alpha Q'(s_t,a_t) && (3) \\ R_k &= \sum_{t=t_k}^{t_{k+1}-1} \gamma^{t-t_i} R(s_t,a_t) && (4) \\ HQ'_k(o_k) &\leftarrow R_k + \gamma^{t_{k+1}-t_k}[(1-\lambda) \max_{o' \in O} HQ_{k+1}(o') \\ &\quad +\lambda HQ'_{k+1}(o_{k+1})] && (5) \\ HQ_k(o_k) &\leftarrow (1-\alpha_{HQ})HQ_k(o_k) + \alpha_{HQ} HQ'_k(o_k) && (6) \end{aligned}$$

ここで，$r_t$ は報酬値，$R(s,a)$ は状態 $s$ で行動 $a$ をとったときに得られる報酬である．$HQ_k(o)$ はサブエージェント $k$ の状態 $o$ に対する HQ 値，$o_i$ はサブエージェント $k$ でそのとき選ばれたサブゴールである．

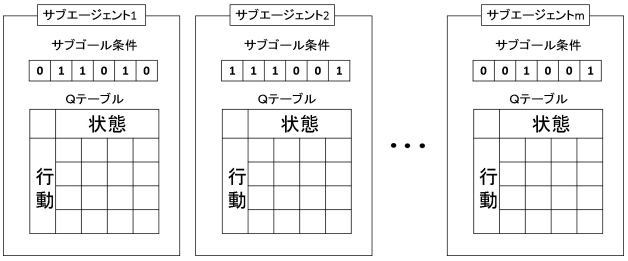



図 2: 遺伝子構造

しかし，HQ テーブルの大きさはサブゴール候補の数だけ必要である．観測情報の増加によりサブゴール候補が指数関数的に増加するため学習効率が著しく低下する．また問題に合わせたサブエージェント数 L の設定が必要となる．サブゴール数は少なすぎるとタスクを達成することができず，多すぎると学習にかかる時間が増加する．そのためタスクがどの程度のサブゴールを必要とするのか予め知っている必要がある．また Q テーブルはサブエージェント 1 つにつき 1 つのみである．これにより，複数のサブゴールを 1 つの Q テーブルで学習するため異なった政策が混在してしまう．また学習が進むにつれあるサブゴールに適した Q テーブルが構築される．そのため，新しいサブゴールを試したときその時の Q テーブルに適していないとタスクを達成できない．よって局所解に陥りやすくなる問題がある．

これらの問題を改善するため，本稿ではサブゴール条件及びその組合せを GA によって自律獲得する手法を提案する．

## 4. 提案手法

エージェントは $m$ 個（$m > 0$）のサブエージェントを持つ．サブエージェントは図 2 に示すように，サブゴール条件と Q テーブルの対（遺伝子対）を持つ．サブゴール条件は観測情報と同じ長さの配列構造である．観測情報と比較し，一致したときサブゴール到達と判定する．サブエージェントの学習には Q-Learning を使用する．行動選択には $\epsilon - greedy$ 法を用い，式（7）により Q 値の更新を行う．

$$\begin{aligned} Q(s_t,a_t) &\leftarrow Q(s_t,a_t) + \alpha[r_{t+1} \\ &\quad +\gamma \max_{a' \in A} Q(s_{t+1},a') - Q(s_t,a_t)] && (7) \end{aligned}$$

### 4.1 サブゴール抽象化

達成不可能なサブゴール条件が与えられたエージェントが多く存在すると，学習効率が低下する．この問題を解決するためドントケアを追加し，サブゴール条件を抽象化する．ドントケアは観測情報が何であっても真をとる．サブゴール判定では，観測情報の配列とサブゴールの配列の各要素で比較されが，サブゴールの要素がドントケアの場合，その要素は一致したとする．これにより達成不可能なサブゴールを減らすだけでなく，サブゴール到達率が上昇し，初期収束を早める効果がある．

### 4.2 スーパーセットカット

学習が進むにつれ，同じサブゴール条件を持ったエージェントが増加する．また，不要なサブエージェントを持つエージェントも生成される．そのため同ルートとなるエージェントの割合が大きくなり，多様性が低下してしまう．多様性維持のため

エージェント $i$ のサブゴール条件の順序付き集合 $S_i$ が多くの包含関係を持った場合に $i$ の適応度を減少させる．まず全てのエージェントの仮の適応度 $F(i)$ を式 9 に従い求める．そして $F$ 値の降順にエージェントの識別子を振り直す．

次に，エージェント $i$ について $S_i \supseteq S_j (0 \leq j < i)$ を満たす $S_j$ の数（$n\_subset$）に応じて $i$ の適応度を減少させる式 (8)．

この処理のねらいは，1) 比較対象を $F$ 値の高いものにすることで仮の適応度 $F(i)$ の高いものを優先に残す，2) サブゴール条件の順序付き集合が $\mathrm{S_i} = \{\mathrm{A,B,C}\}, \mathrm{S_j} = \{\mathrm{A,C}\}$,$F(i) > F(j)$ となるような場合においては，サブゴール B が適応度を上げる要因である可能性があるため．$\mathrm{S_i}$ の適応度を減少させない，ことにある．

### 4.3 適応度

エージェント $i$ の適応度 $F'(i)$ を式（8）で定義する．

$$F'(i) = \frac{F(i)}{2^{n\_subset_i}} \tag{8}$$

$$F(i) = \begin{cases} \frac{(\mathrm{R}+(\mathrm{Maxstep}-step_i)/\mathrm{b})}{don't_i+1} & (\text{learned}) \\ \mathrm{r} + goal_i/\mathrm{a} & (\text{unlearned}) \end{cases} \tag{9}$$

ここで，r は最低報酬値，R はゴール報酬値，$goal_i$ はゴール回数，Maxstep は最大ステップ数，$step_i$ はステップ数，$don't_i$ はドントケアの総数，a と b は重みを表す．学習終了後 greedy 法で行動を選択し 1 試行する．このときゴールできたエージェントを学習完了（learned），ゴールできなかったエージェントを学習未完了（unlearned）とする．

### 4.4 交叉

サブゴール条件と，遺伝子対の組合せのそれぞれについて交叉を行う．サブゴール条件の交叉（交叉 1）は，エージェント $i, j$ のサブエージェントのサブゴール条件 $\mathrm{S_i}$ と $\mathrm{S_j}$ を一様交叉をする．しかしエージェントの持つサブエージェントの数が異なるため，$|\mathrm{S_i}| = |\mathrm{S_j}|$ となるように $\mathrm{S_j}$ を調整する．この交叉により生成されたエージェントのサブエージェントの Q テーブルは初期化する．

遺伝子対の組合せの交叉（交叉 2）は，エージェント $i, j$ のサブエージェントの順序付き集合 $\mathrm{S_i}$ と $\mathrm{S_j}$ を一点交叉する．エージェント $i$ の前半部分 $\mathrm{S_i[k]}$ $(0 < k \leq x_i)$ とのエージェント $j$ の後半部分 $\mathrm{S_j[h]}$ $(x_j \leq h < |S_j|)$ を結合する．$\mathrm{x_i}$ は $0 < x_i \leq |\mathrm{S_j}|$ の範囲でランダムに選択する．これによりサブエージェントの数が動的に変化する．この交叉により生成されたエージェントのサブエージェントは親個体の Q テーブルをそれぞれ引き継ぐ．

### 4.5 世代交代

次世代に残す個体はエリート個体，交叉 1 および交叉 2 によって生成された個体である．これらの個体の構成比は，エリート個体の割合を増やすと多様性を保持しやすい反面学習効率が低下する．交叉 1 により生成された個体の割合を増やすと新しいサブゴールの発見率が増加する．交叉 2 により生成された個体の割合を増やすと前世代に近しいルートの探索率が増加する．

## 5. 実験

### 5.1 実験 1，比較実験

Wiering ら [2] が作成した 12 × 12 迷路（図 3）を利用して評価実験を行った．エージェントは 8 近傍の壁と道を観測する．行動は「上」「下」「左」「右」の 4 通り．壁に向かって進む場合はその場に留まるが 1 ステップとして数える．提案手法のパラメータは学習率 $\alpha$=0.9，割引率 $\gamma$=0.7．次世代に残す個体率はエリート保存が 0.3，交叉 1 が 0.2，交叉 2 が 0.5．世代数 50，エージェント数 50，エージェントの Q-Learning の試行数 300，最大ステップ数 100．初期サブエージェント数 4 とした．HQ-Learning のパラメータは Q テーブルの学習率 $\alpha$=0.05，割引率 $\gamma$=0.9，重み $\lambda$=0.9，HQ テーブルの学習率 $\alpha_{HQ}$=0.2，温度 T=1，サブエージェント数 L=4，$P_{max}$ は最初の試行が 0.4 で，最後の試行までに線形的に 0.8 まで上昇させる．試行数は提案手法に合わせるため世代数×エージェント数×試行数である 750000 とした．*1最大ステップ数は，提案手法と同じ 100 ステップ（HQ-Learning(100step)）と Wiering らが行った 1000 ステップ（HQ-Learning(1000step)）の 2 つ実験した．

図 4 に実験結果を示す．同図は試行数に対する平均最短経路長の変化を示す．ただし提案手法では GA を用いるため，HQ-Learning の試行数に対応した世代数を付記した．提案手法では十分に学習が進み，経路長が最適解である 28 ステップに収束した．一方で，HQ-Learning(100) の平均最短経路長が約 70，HQ-Learning(1000) が約 50 となった．これはゴール出来なかったり局所解に陥ったエージェントが多く存在したためである．提案手法ではこの問題をサブゴール抽象化と GA によって解決できたと考えられる．サブゴール抽象化によりゴールできないエージェントを減らし，初期収束を速めた．GA により今までの学習結果を利用し効率よく学習した．また，提案手法ではサブゴール数が必要最低限である 2 となった．これはサブゴール条件に包含関係を持つエージェントを淘汰することで，冗長な表現を含むエージェントを減らすことができたためと考えられる．

### 5.2 実験 2，環境変化

環境変化の実験として 22 × 22 迷路（図 5(a)）を作成した．まず 100 世代学習し，その後 (12,6) を壁に変え再び 100 世代学習する．その他パラメータは実験 1 と同じものを使用する．HQ-Learning は変化のある環境には適さないため実験から除外した．

図 6 に実験結果を示す．同図は世代数に対するタスク達成率を示す．greedy 法を行い最大ステップ数内にゴールできた場合にタスク達成とする．100 世代で環境変化が起き達成率が減少するが，約半数が 1 世代内でタスク達成し他エージェントも 40 世代内に達成した．エージェントが複数の経路を保持することで経路が塞がれても即時に対応ができる場合が多い．また今回の例では経路の前半部分に変化はないため，交叉 2 により一部の経路を再利用することで復帰も早くなったと考えられる．

図 5(b) に実験 2 の 100 世代での得られた経路の例を示す．同図は 50 エージェントのうち，タスク達成した $m$ 個の個体の獲得された経路を表している．赤色が濃いほど多くのエージェントが通った経路となる．いくつかの経路を保持できていることが分かる．今回の環境変化を回避する経路が既に学習できているため，環境変化に即時に対応できた．しかし，ある経路に大きく偏っていることが分かる．スーパーセットカットにより多様性維持を試みたが未だに同経路を保持している．また最適解である 40 ステップを達成することができなかった．この最適解を満たすためには非常に限られたサブゴールの組み合わせを発見し，またそれに合わせた Q 値を学習する必要があ

*1 提案手法の 1 世代が HQ-Learning の 1500 試行に相当する．

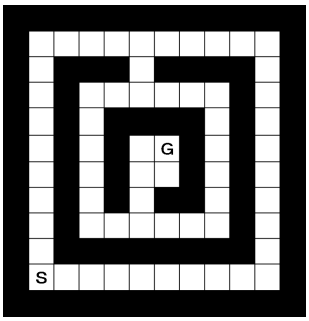
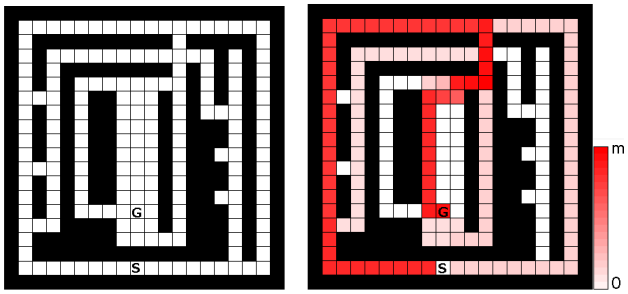



図 3: 12 × 12 迷路




(a) 22 × 22 迷路　　(b) 獲得経路例

図 5: 22 × 22 迷路

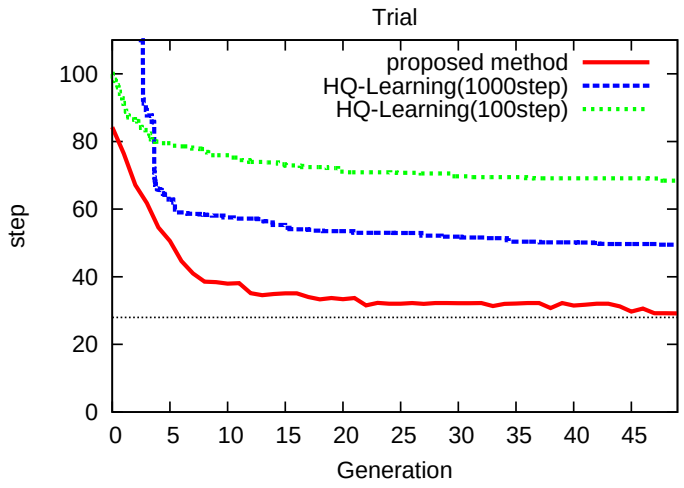



図 4: 実験結果 50 回平均

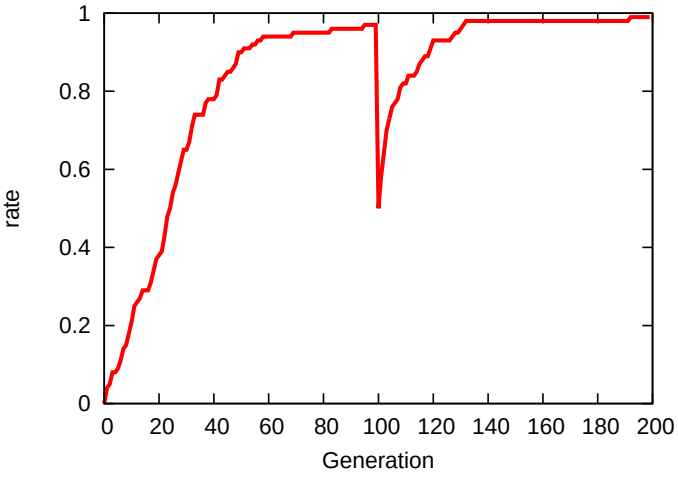



図 6: 実験結果 100 回平均

る．そのためこの経路を発見することは困難だった．サブゴールをエージェント自身の行動によって得られた観測情報を元に生成することで問題が緩和できるのではないかと考える．

## 6. 考察

HQ-Learning が一括学習，提案手法が逐次学習であるが，収束の早さが上回った．これはサブゴール抽象化によるものと考えられる．これは HQ-Learning にも適用可能だが，学習対象が増加しサブゴールのドントケアが多く残るといった問題がある．ドントケアが多く残ると，部分観測の度合いが増すため，サブゴール到達を誤って判定しやすくなりノイズに弱くなる．

局所解は GA においても大きな問題であるがスーパーセットカットによる多様性維持により，実験 1 においては十分な結果が得られた．しかし実験 2 に示すように局所解を完全には回避できなかった．

今回はサブゴール条件を抽象化する機能を与えた．しかし提案手法は 1）サブゴール判定ができる 2）交叉ができる の 2 つが満たされれば学習可能である．サブゴール条件を木構造にした遺伝的プログラミング等も可能である．

## 7. おわりに

本稿では POMDP に対して，サブゴール条件及びその組合せを GA で自律獲得する手法を提案した．今後は，観測情報を拡大した実験を行うことで手法の有効性を確認していく．今後の展望として，スタート地点のランダム化が挙げられる．今回の実験例ではスタートとゴールが固定されていたため，行動系列のみを学習すればよかった．迷路全域での最適な行動を十分に学習できていないため，スタート地点をランダムに設定し，より汎用的な環境下に対応可能なシステムの構築を目指す．

## 謝辞

本研究は，一部，文部科学省科学研究費補助金（課題番号 25280100，および，25540146）の助成により行われた

## 参考文献

[1] S. D. WHITEHEAD. Learning to perceive and act by trial and error. *Machine Learning*, Vol. 7, pp. 45–83, 1991.

[2] M. WIERING. HQ-learning. *Adaptive Behavior*, Vol. 6, No. 2, pp. 219–246, 1998.

[3] 和光宮崎, 幸代荒井, 重信小林. 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, jan 1999.

[4] C. J. C. H. WATKINS. Technical note : Q-learning. *Machine Learning*, Vol. 8, No. 3, pp. 279–292, 1992.