論　文

状態空間の部分的高次元化法によるマルチエージェント強化学習

藤田　和幸[†]　　松尾　啓志[†a)]

Multi-Agent Reinforcement Learning with the Partly High-dimensional State Space

Kazuyuki FUJITA[†] and Hiroshi MATSUO[†a)]

あらまし　マルチエージェントシステムの設計手法の一つとして，マルチエージェント強化学習がある．マルチエージェント強化学習において，エージェントは自分以外のエージェントも環境の一部として観測する．そのため，エージェント数の増加に伴い状態空間が指数的に増加し（次元の呪），学習速度を著しく低下させるという問題が生じる．また，学習に必要なメモリ量も膨大となる．この問題を解決する手法として提案された Modular Q-learning には，知覚の不完全性により学習性能が低下する問題がある．本研究では状態空間の部分的高次元化手法を用いて Modular Q-learning の学習性能を改善する手法 HMQL を提案する．

キーワード　マルチエージェント，強化学習，Modular Q-learning，状態空間

1. まえがき

近年，分散協調システムの一つとしてマルチエージェントシステムが注目されている．マルチエージェントシステムは，自律的に行動する複数のエージェントから構成されるため，一般に中央制御を必要としない．マルチエージェントシステムを構成するエージェントの設計方法の一つに強化学習 [1] がある．強化学習は教師なし学習の一種であり，システムの設計者が環境に応じた設計を施す必要がないため，大規模で複雑なシステムを設計する場合などに非常に有効な手法であるといえる．マルチエージェント強化学習に関する研究は，実問題への適用 [3] [4]，社会シミュレーションへの応用 [5]，実ロボットを用いた協調行動獲得 [6] [7] など多岐に渡る．しかし，マルチエージェント強化学習には，荒井 [2] が示しているように不完全知覚問題，同時学習問題，報酬配分問題などの問題がある．また，これら以外の問題として次元の呪がある．

本論文では，次元の呪について議論する．シングルエージェント環境においても次元の呪は存在する．実環境に強化学習を適用する場合，とりわけロボット制御などでは行動出力が連続値であり，状態空間が連続値で与えられる．そのため，連続値を離散化して学習を行う方法では，行動空間，状態空間が爆発する．次元の呪が生じる環境では，エージェントが学習空間を探索するために多くの時間を必要とし，学習速度を著しく低下させてしまう．シングルエージェント環境では，連続的な行動空間に対して Actor-Critic 法が提案されており，連続的な状態空間に対しては，SVM(Support Vector Machine) を用いた状態一般化法 [8]，状態空間の階層化 [9] などがある．

シングルエージェント環境においても，強化学習を実環境に用いる場合には学習速度の低下が問題とされている．したがって，シングルエージェント環境下での学習よりも多くの学習時間を必要とするマルチエージェント強化学習では，学習速度の低下は一層重要な問題といえる．

さらに，マルチエージェント強化学習では，エージェント数の増加が次元の呪を招く要因としてさらに重要な問題となる．なぜなら，全てのエージェントを環境の一部として状態空間を構成することで，エージェント数の増加に伴い状態空間は指数的に増大し，状態空間が爆発してしまうからである．高橋ら [9] の手法は，下位層で状態空間を幾つかの小さな領域に分割し，各領域において独立して学習を行うことで膨大な状態空間に対応している．また，高位層では下位層学習器間の移動を定義しており，下位層において目標状態への

[†] 名古屋工業大学電気情報工学科，名古屋市
Department of Electrical and Computer Engineering, Nagoya Institute of Technology, Gokiso-cho, Showa-ku, Nagoya, 466-8555, Japan
a) E-mail: matsuo@nitech.ac.jp

到達が不可能な場合には，高位層の学習器を用いて目標状態に近い下位層の学習器への移動を行うというものである．しかし，状態遷移に他エージェントの行動という不確定な要素が含まれるマルチエージェント環境において，抽象化された高次元空間における状態遷移を定義することは非常に困難である．また，Fuzzy システムを用いた状態数の削減手法 [10] があるが，個々のエージェントの状態数が減少するのみで，依然として学習の対象となる状態空間はエージェント数の増加に伴い指数的に増大する．その他にも，知覚情報を制限することで状態空間を削減する手法 [11] がある．

マルチエージェント強化学習における次元の呪を解決する手法として Modular Q-learning [12] がある．この手法は，2 体エージェントから構成される部分状態を用いるため，状態空間の大きさがエージェント数に影響されない．しかし，部分状態のみを観測するため，不完全知覚による学習性能の低下を招く．そこで本研究では，学習性能低下の要因となる状態を高次元化することで不完全知覚を取り除き，Modular Q-learning の学習性能を改善する手法 HMQL（Hybrid Modular Q-Learning）を提案する．

以下，2 章では次元の呪と Modular Q-learning の問題点について述べる．3 章では，提案手法について説明し，4 章で追跡問題を用いて提案手法の性能評価を行う．

2. 次元の呪

エージェント i がセンシング可能なエージェント j の状態を s_j，エージェント数を N としたとき，全てのエージェントを環境の一部として状態空間を構成する場合にエージェント k が学習に用いる環境状態 S_k は，全エージェントの状態の集合 $< s_1, \cdots, s_N >$ から成る．

したがって，状態空間の大きさ $|S_k|$ は $|s|^N$ となり，エージェント数が増加すると $|S_k|$ は指数的に増大する．強化学習において状態空間の増大は学習速度を急激に低下させ，また膨大なメモリ量を必要とさせる．

2.1 Modular Q-learning

Modular Q-learning は状態空間の爆発を防ぐ手法として Ono らによって提案された手法である．Modular Q-learning では，自分と他の 1 体のエージェントから構成される部分状態空間を用いる．そのため状態空間の大きさは，エージェントの数に関わらず常に $|s|^2$ となり，状態空間の爆発を防ぐことが可能となる．

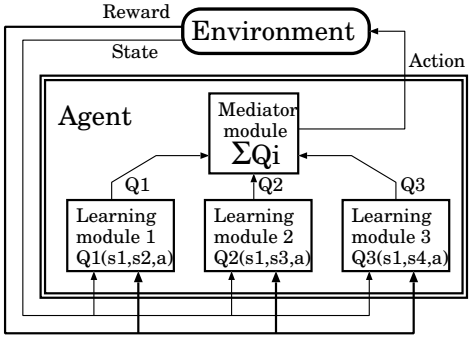


図 1　Modular Q-learning におけるエージェントの構成
Fig. 1　Structure of an agent in Modular Q-learning

エージェントが N 体存在する場合，部分状態空間は $N-1$ 個存在し，それぞれに対して一つずつ学習器が割り当てられている．図 1 にエージェントが 4 体の場合のエージェント 1 の構成を示す．ここで，エージェント 1 は 4 体の内の 1 体であり，3 つの部分状態空間 $< s_1, s_2 >, < s_1, s_3 >, < s_1, s_4 >$ を学習に用いるエージェントである．各学習器は Q-learning を行い，Q 値を行動選択器へと渡す．Q-learning では，状態 s_t で行動 a_t を選択して状態 s_{t+1} へと遷移した場合，選択したルールの Q 値 $Q(s_t, a_t)$ は次の式 (1) で更新される．

$$\begin{aligned} Q(s_t, a_t) \leftarrow\ & (1-\alpha)Q(s_t, a_t) \\ & + \alpha(r_t + \gamma \max_a Q(s_{t+1}, a)) \end{aligned} \tag{1}$$

ここで，α は学習率 $(0 < \alpha < 1)$，γ は割引率 $(0 < \gamma < 1)$，r_t は報酬を表している．式 (2) に示すように行動選択器は受け取った Q 値の合計値がより大きな値となる行動を優先して選択する [13]．ここで，a は行動，A は行動の集合，$\arg\max_{a \in A} Q$ は Q を最大とするような引数 a を意味する．また，$S_{k,i}$ はエージェント k とそれ以外のエージェント i から構成される部分状態を表している．

$$\arg\max_{a \in A} \sum_{i=1}^{i=N-1} Q^{k,i}(S_{k,i}, a) \tag{2}$$

しかし，部分状態のみを観測することで不完全知覚を招き，学習性能が低下する欠点がある．図 2 に性能低下の例を示す．

図 2 では，状態 S_0 と状態 S_5 において同一の部分状態 $S_{1,2}$ を観測する．状態 S_5 において最適な行動

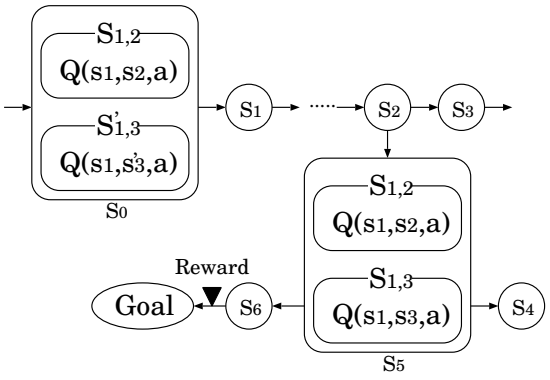


図 2　Modular Q-learning における学習性能低下の例：状態 S_0 と状態 S_5 では同一の部分状態 $S_{1,2}$ を観測する
Fig. 2　An example that Modular Q-learning's performance down.

は状態 S_6 へ遷移する行動であり，その行動の Q 値 $Q(s_1, s_2, a)$ が状態 S_5 において最大となっていると仮定する．状態 S_0 における $Q(s_1, s_2, a)$ は状態 S_5 における $Q(s_1, s_2, a)$ と同一のものであり，また状態 S_0 はゴール状態から遠いため，$Q(s_1, s_3^{'}, a)$ は $Q(s_1, s_2, a)$ に比べて非常に小さい．つまり状態 S_0 においても，状態 S_5 における最適行動と同じ行動を選択する可能性が高く，この例では同じ行動を選択した場合を考える．

状態 S_0 から遷移可能な状態は状態 S_0 と同様にゴールから遠く，Q 値も小さい．つまり，Q 学習において状態遷移先の価値として用いられる $\max_a Q(s, a)$ の値は $Q(s_1, s_2, a)$ より小さい．このため，学習を行うことで $Q(s_1, s_2, a)$ の値は減少する．これは，状態 S_5 において最適な行動を選択する可能性が低下することを意味し，状態 S_6 ではなくゴール状態から遠い状態 S_4 へと遷移する可能性が高くなる．また，状態 S_2 から状態 S_5 へと遷移した場合，学習に用いられる $\max_a Q(s, a) = Q(s_1, s_2, a)$ の値が本来の値よりも小さいため，状態 S_5 へと遷移する行動の Q 値も本来の値よりも小さな値となり，状態 S_3 へと遷移する可能性が高くなる．状態 S_3，状態 S_4 への遷移は最適な政策ではないため，学習性能が低下すると考える．

3. 提案手法：HMQL（Hybrid Modular Q-Learning）

Modular Q-learning では部分状態のみを観測しているため，不完全知覚が生じ学習性能が低下した．そこで，図 2 の状態 S_5 のようにゴール状態に近く，Q 値が低下すると学習性能に大きな影響を与える状態では，部分状態 $S_{1,2}, S_{1,3}$ を用いるのではなく，それらの状態を高次元化して得られる状態 $S_{1,2,3} = \{s_1, s_2, s_3\}$ を用いることで学習性能の低下を防ぐことができると考えられる．なぜなら，状態 $S_{1,2,3}$ を用いることは，状態 S_5 では $Q(s_1, s_2, a), Q(s_1, s_3, a)$ ではなく $Q(s_1, s_2, s_3, a)$ にしたがって行動を選択することを意味し，状態 S_0 からの遷移によって $Q(s_1, s_2, a)$ の値が低下しても，状態 S_5 における行動選択は影響を受けないためである．また，状態 S_2 から状態 S_5 へと遷移した場合も，$\max_a Q(s, a) = Q(s_1, s_2, s_3, a)$ を用いることで，状態 S_5 へと遷移する行動の Q 値の低下を防ぐことができる．状態 S_0 を高次元化しても同様のことが言えるが，そもそもゴールから遠い状態は価値が低く，遷移する確率も小さい．したがって，そのような状態を高次元化しても，状態 S_5 を高次元化した場合ほど性能改善の効果は期待できない．また，多くの状態を高次元化することによって扱う状態数が増えすぎると，Modular Q-learning の利点である速い学習速度が失われてしまう．

そこで本手法では，図 2 における状態 S_5 のような状態のみを高次元化し，不完全知覚を取り除くことで Modular Q-learning の学習性能を改善する．ただし本手法では，状態価値の高い状態が価値の低い状態にくらべて，報酬獲得に寄与する度合いが大きいことを仮定している．すなわち，Q-learning や Profit Sharing のようにゴールに近い状態に対して高い価値を割り当てる強化学習が有効な問題に対して，本手法は有効であると考える．

3.1 高次元化法

高次元化を行うためには，まず現在の状態が図 2 の状態 S_5 のような状態にあるか否かを決定する必要がある．また上述したように，本手法では，可能な限り少ない高次元状態を用いることで，学習速度を損なわない効率的な性能改善を目的としている．したがって，状態 S_0 と状態 S_5 を識別する必要がある．しかし，部分状態を観測して得られる情報だけでは決定することはできない．そこで，部分状態の価値を表す状態価値関数 V_i を用いて，状態の識別を行う手法を提案する．自分の状態 s_{self} と他エージェント i の状態 s_i から構成される部分状態の価値 $V_i(s_{self}, s_i)$ の学習式を式 (3) で与える．

$$V_i(s_{self}^{t-1}, s_i^{t-1}) \leftarrow V_i(s_{self}^{t-1}, s_i^{t-1})$$

$$+\beta(\max(r_t, \gamma V_i(s^t_{self}, s^t_i)) - V_i(s^{t-1}_{self}, s^{t-1}_i)) \quad (3)$$

ここで，β は学習率 $(0 < \beta < 1)$，γ は割引率 $(0 < \gamma < 1)$ である．状態価値関数 V_i の値は，報酬が与えられるか，より価値の高い状態へと遷移した場合に増加する．つまり，ゴールに近い状態ほど価値 V の値が大きく，ゴールから遠い状態ほど価値 V の値は小さくなる．状態 S_5 はゴール状態に近いため $V(s_1, s_2), V(s_1, s_3)$ の値は大きくなる．一方，状態 S_0 はゴール状態から遠く $V(s_1, s_3^{'})$ の値は小さい．状態 S_0 から状態 S_1 へと遷移することで $V(s_1, s_2)$ の値は減少するが，少なくとも $V(s_1, s_3^{'})$ よりは大きくなる．そこで，$V(s_1, s_2) > \eta$, $V(s_1, s_3) > \eta$, $V(s_1, s_3^{'}) < \eta$ となる閾値 η を設け，状態を構成する部分状態の価値が共に閾値を超える状態を状態 S_5 として認識し，状態 S_0 と区別することが可能となる．このようにして状態 S_5 にあると判定された状態を高次元化する．価値 V の値は学習し続けるため，学習途中で減少し $V(s_1, s_2) < \eta$ となる可能性もある．本手法では，一度高次元化された状態は性能改善に重要な状態であると考え，以後 V が低下しても高次元状態として扱う．ただし，環境が変化する場合は考慮していない．

閾値 η は式 (4) で与える．

$$\eta = \gamma^{\lambda} \cdot Reward \quad (4)$$

λ はパラメータ，γ は状態価値関数 V の学習式に用いた割引率，$Reward$ は環境から与えられる報酬を表す．λ の値を無限大に近づけると閾値 η の値は 0 となる．そのため，全ての状態が高次元化され，学習アルゴリズムは Q-learning と等しくなる．したがって，λ の値を大きくすることで学習性能は大きく改善されるが，学習速度が低下する．また，λ の値を非常に小さな値に近づけると閾値 η の値は無限大となる．そのため，高次元化される状態は存在せず，学習アルゴリズムは Modular Q-learning と等しくなる．つまり，λ の値を小さくすることで学習速度は速くなるが，学習性能が低下する．

本手法は，この λ を適切な値に設定することで，十分な学習速度を保ったまま，Modular Q-learning の学習性能を改善することが可能である．

3.2 エージェントの構成

図 3 にエージェントの構成を示す．

- Check module:

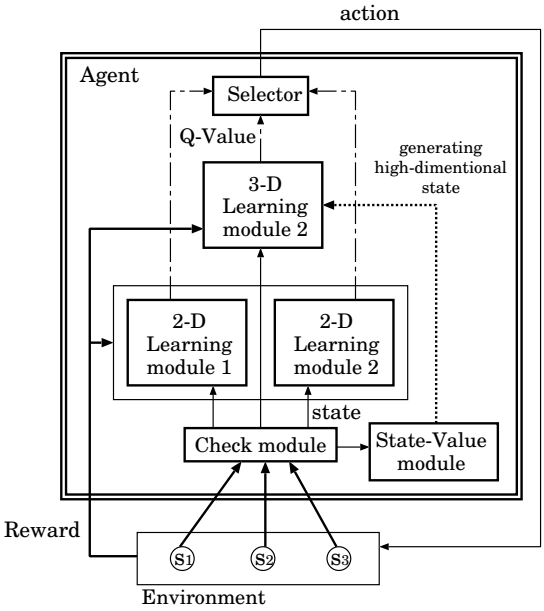


図 3　エージェントの構成
Fig. 3　Structure of an agent

次元数に応じた学習器を選択するモジュール．各状態ベクトルの次元数を記憶しており，観測した状態ベクトル (s_1, s_2, s_3) が 2 次元状態として扱われるならば，2 次元学習器に部分状態 $\{s_1, s_2\}$ と $\{s_1, s_3\}$ を渡し，3 次元状態として扱われるならば，状態 $\{s_1, s_2, s_3\}$ を 3 次元学習器に渡す．また，2 次元部分状態を State-Value module に渡す．

- State-Value module:

高次元化の判定と高次元状態の生成を行うモジュール．Check module から 2 次元部分状態を受け取り，閾値 η を用いて高次元化の判定を行う．また，高次元化の条件を満たしているならば高次元状態を生成する．

- N 次元学習器 (Learning module)

N 次元状態の学習を行うモジュール．学習には長行ら [14] によって提案された，他エージェントの政策推定を利用した Q-learning を用いる．学習を行う学習器は Check module によって選択された学習器のみで，選択された学習器は行動選択に必要となる Q 値を行動選択器に渡す．また，報酬も選択された学習器のみに与えられる．

- 行動選択器 (Selector)

各学習器から受け取った Q 値を用いて行動選択を行うモジュール．受け取った Q 値を行動ごとに合計し，

その合計値がより大きな値を持つ行動を優先した戦略を取る．

3.3 学習アルゴリズム

本手法の学習アルゴリズムを以下に示す．

（1） 各学習モジュールが持つ Q 値の値を初期値に設定する．また，Check module が記憶している全状態ベクトルの次元数を，本手法で扱う最小次元（2 次元）に設定する．

（2） 環境を観測する．

（3） 観測して得られた知覚をもとに，Check module を用いて高次元状態が存在するか調べる．また，State-Value module を用いて高次元化条件に基づいた状態の高次元化を行う．そして，現在の状況に対応した次元の学習モジュールを選択し状態を与える．

（4） 選択された学習モジュールは Q 値を行動選択器に渡す．

（5） Q 値を受け取った行動選択器は Q 値の合計値をもとに行動を選択する．

（6） 環境を観測し，与えられた報酬に基づいて Q 値・状態価値を学習する．

（7） ゴール状態なら終了．そうでなければ (3) へ．

4. 評価実験

4.1 追跡問題

提案手法の性能評価を行うために，追跡問題を用いる．追跡問題とは，ハンターが獲物を追跡し捕獲する問題である．本研究では複数のハンターと獲物が存在する環境において，ハンターをエージェントと見なし，エージェントに獲物捕獲行動を学習させる．以下に，本研究で用いた追跡問題の環境設定を示す（長行らが評価実験に用いた問題と同様の問題設定である）．

- 2 次元トーラス状グリッド（5x5 もしくは 9x9）に，3 体のハンターと 1 体の獲物が存在する．
- ハンターと獲物は，各時間ステップに同時に行動を決定する．また，ハンターが選択可能な行動は 5 通りあり，上，下，左，右の隣接するグリッドへの移動，現在の位置に停止である．獲物が選択可能な行動は 3 通りあり，上へ移動 (40%)，右へ移動 (40%)，現在の位置に停止 (20%) である．また，括弧内の数字は行動選択確率を表す．
- 獲物は学習を行わず，行動選択確率は一定とする．
- 獲物とハンターが同じグリッドに移動することを許可する．

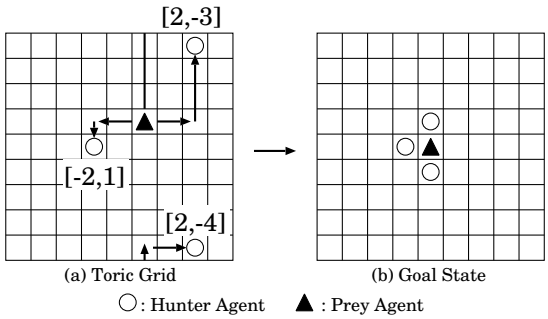


図 4 追跡問題：(a) トーラス状のグリッド，矢印で示すように相対位置を計算する．(b) ゴール状態

Fig. 4 pursuit domain:(a)toric grid,The arrow explain relative position. (b)goal state

- 初期状態から獲物を捕獲する（ゴール状態）までを 1 エピソードとする．獲物とハンターの初期配置はランダムに決定する．また，ゴール状態は図 4(b) に示すように，ハンターが異なる 3 方向から獲物に隣接した状態とする．
- エージェントの状態は，獲物との相対位置で表現する．例えば，図 4 に示すように，獲物との相対位置は $[-2,1],[2,-3],[2,-4]$ のように表現する．
- 獲物捕獲した場合，エージェントに報酬 1.0 を与え，それ以外の場合には報酬-0.05 を与える．

4.2 政策推定を用いた学習

本手法では，各学習器における学習は長行らによって提案された，他エージェントの政策推定を利用した Q-learning を用いる．他エージェントを環境の一部として観測するマルチエージェント環境では，他エージェントの政策変化に伴い環境の状態遷移確率も変化する．そのため，マルチエージェント環境は MDP（マルコフ決定過程）環境としてモデル化することができない．学習時に遷移先の Q 値を用いる Q-learning をそのままマルチエージェント環境に適用することは合理性に欠ける．長行らは他エージェントの政策推定を行うことで，他エージェントの政策変化により状態遷移確率が時間と共に変化する環境においても有効な政策を獲得している．

以下に長行らの手法の概略を示す．エージェント k は式 (5) を用いて，他エージェント o が状態 S_{t-1} において行動 a^o_{t-1} を選択する確率の推定値 $I^k(S_{t-1}, a^0_{t-1})$ を更新する．

全ての行動 $a \in A$ に対して

$$I^k(S_{t-1}, a) \leftarrow (1-\theta) I^k(S_{t-1}, a)$$

$$+\begin{cases} \theta & (a = a^o_{t-1}) \\ 0 & (otherwise) \end{cases} \quad (5)$$

また，時刻 t においてエージェント k は次の式 (6) で与えられる $\bar{Q}^k(S_t, a^k)$ の値をもとに行動選択を行う．

$$\bar{Q}^k(S_t, a^k) = \sum_{a^o} I(S_t, a^o) Q^k(S_t, a^k, a^o) \quad (6)$$

実際に学習されるのは Q^k で，学習式は式 (7) で与えられる．

$$Q^k(S_t, a^k_t, a^o_t) \leftarrow (1-\alpha) Q^k(S_t, a^k_t, a^o_t) + \alpha(r^k_{t+1} + \gamma \max_{a^k} \bar{Q}^k(S_{t+1}, a^k)) \quad (7)$$

長行らの手法は 2 体エージェント確率ゲームのみに適用されている．しかし，本手法では 3 体以上のエージェントも扱うため，式 (6)，式 (7) の自然な拡張として以下の式を用いた．

$$\bar{Q}^k(S_t, a^k) = \sum_{a^{o1}} \cdots \sum_{a^{oN-1}} I(S_t, a^o_1) Q^k(S_t, a^k, a^{o1}, \cdots, a^{oN-1}) \quad (8)$$

$$Q^k(S_t, a^k_t, a^{o1}_t, \cdots, a^{oN-1}_t) \leftarrow (1-\alpha) Q^k(S_t, a^k_t, a^{o1}_t, \cdots, a^{oN-1}_t) + \alpha(\max(r^k_{t+1}, \gamma \max_{a^k} \bar{Q}^k(S_{t+1}, a^k))) \quad (9)$$

提案手法の有効性を明らかにするために，今回比較手法として用いる Modular Q-learning，Q-learning にも同様の政策推定を用いた．

以上の環境で，獲物捕獲に要するステップ数・状態空間の大きさについて，Modular Q-learning，Q-learning との比較を行う．また，各手法における行動選択にはボルツマン選択（式 (10)）を用いた．

$$\pi(s_t, a) = \frac{e^{Q(s_t,a)/T}}{\sum_{b \in A} e^{Q(s_t,b)/T}} \quad (10)$$

ここで，$\pi(s_t, a)$ は状態 s_t で行動 a を選択する確率で，T は温度パラメータである．

4.3 実験結果

a） 5x5 グリッド環境

図 5 に，提案手法，Modular Q-learning，Q-learning を用いた実験結果を示す．横軸はエピソード数，縦軸は獲物捕獲に要したステップ数を表している．提案手法，Modular Q-learning，Q-learning の結果は共に 100 試行の平均値である．また，図 5 の値は 10 エピソード毎の移動平均である．各学習法におけるパラメータの値は，学習率 α は $0.3 \times 0.998849^{epi}$，割引率 $= 0.9$，温度パラメータは $0.1 \times 0.998849^{epi}$ とした．ここで，epi はエピソード数を表し，$0.998849^{10000} \approx 0.1$ となるように温度パラメータの減衰率を決定した（長行らのパラメータ決定法と同様である）．また，閾値パラメータ λ の値は 4.0 とした．λ の値は予備実験によって得られた最適な値であり，この値はエピソードを通して固定とする．

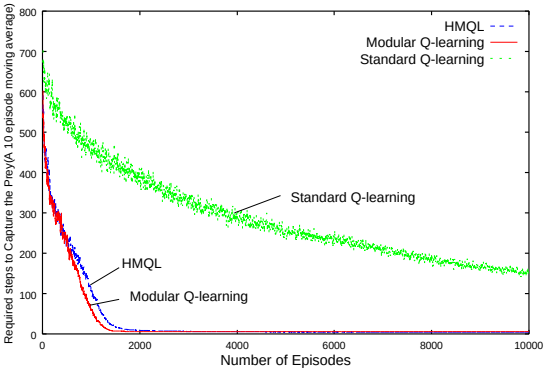


図 5 5x5 グリッド環境における実験結果
Fig. 5 An experimental result in 5x5 grids

本環境での Q-learning における状態空間の大きさは 15625 状態であり非常に大きい．そのため 1 エピソードで学習の対象となる状態数が，Modular Q-learning および提案手法に比べて相対的に少ないため，学習速度が遅く，10000 エピソード学習を行っても獲物捕獲に 150 ステップ程必要としている．一方，提案手法，Modular Q-learning における状態空間の大きさはそれぞれ 1100 状態，625 状態であり，Q-learning に比べて非常に小さく，図 5 が示すように学習速度も Q-learning に比べて非常に速い．

次に，図 6 に提案手法と Modular Q-learning が獲物捕獲に要したステップ数を 0 ～ 20 ステップの範囲で示す．提案手法では，高次元化手法によって扱う状態数が増えるため，Modular Q-learning よりも学習速度が若干低下している．しかし，Modular Q-learning のステップ数がほぼ収束したと思われる 3500 エピソード付近で，Modular Q-learning と提案手法のステップ数が同じ値となっている．さらにその後のエピソードでは，提案手法のステップ数が Modular Q-learning

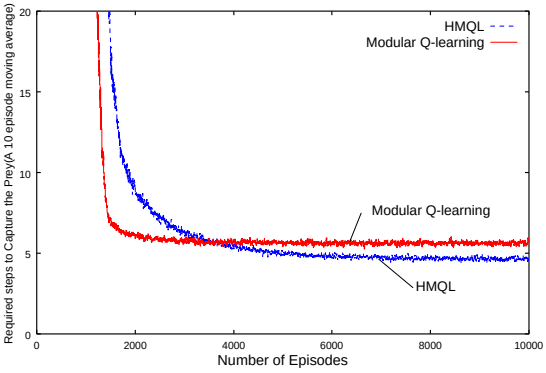


図 6　5x5 グリッド環境における実験結果：HMQL と Modular Q-learning の結果のみを表示
Fig. 6　An experimental result in 5x5 grids:A proposal method and Modular Q-learning.

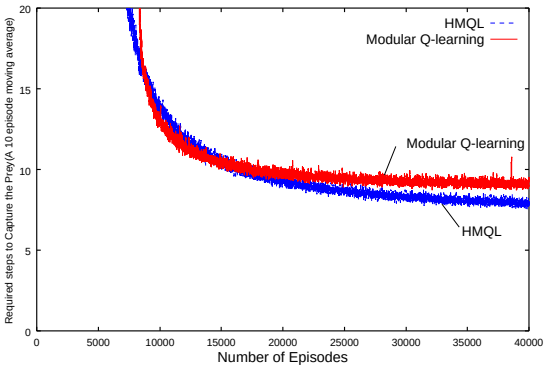


図 7　9x9 グリッド環境における実験結果
Fig. 7　An experimental result in 9x9 grids

のステップ数よりも小さな値となっており，試行の終了時では Modular Q-learning のステップ数を約 17%改善可能であった．このことから，高次元状態を用いて不完全知覚を取り除くことにより，Modular Q-learning の学習性能を改善可能であることが確認できた．

b）　9x9 グリッド環境

グリッドの大きさを 9x9 として実験を行った．9x9 グリッドでは Q-learning における状態空間の大きさは 531441 状態と非常に大きい．また，5x5 グリッド環境の実験において明らかに Q-learning の学習速度は遅く，9x9 グリッド環境においても同様の結果になると考えられる．また，学習に要するメモリ量も膨大となるため，9x9 グリッド環境においては Modular Q-learning のみと比較を行った．閾値パラメータ λ の値は 7.0 とした．図 7 に提案手法と Modular Q-learning を用いた実験結果を示す．

5x5 グリッド環境における実験結果と同様に，学習途中から提案手法のステップ数が Modular Q-learning のステップ数よりも小さな値となっている．また，Modular Q-learning，提案手法における状態空間の大きさはそれぞれ 6561 状態, 7300 状態であり，提案手法の状態空間の方が大きい．しかし，図 7 が示すように，提案手法の学習速度と Modular Q-learning の学習速度には 5x5 グリッド環境の時ほど差はない．この理由として以下のことが考えられる．

Q-learning において，報酬はゴール近傍の状態からより遠くの状態へと伝播する．また，割引率によってより遠くの状態ほど伝播する報酬は少なくなる．つまり，割引率があまりに小さな値である場合や，ゴールに到達するまでに非常に多くのステップ数を必要とする場合には，ゴールから遠い状態における Q 値はほとんど学習されない．ただし，割引率を適切な値に設定することでこのような問題は生じない．しかし Modular Q-learning では，これ以外の要因でゴールから遠い状態に伝播する報酬が減少し，学習速度が低下してしまう．例えば，図 8 のような状況を想定する．獲物が以後停止し続けると仮定した場合，丸で囲まれた 2 体のエージェントと獲物からなる部分状態（以下，部分状態 S）において，各エージェントは現在位置に停止する行動が最適な行動となる．また，残り 1 体のエージェントは単純に獲物に近づく行動が最適な行動となる．本来最適な行動を選択した場合，その Q 値は増加しなければいけない．しかし，この状況において左下のエージェントが最適な行動（5 ステップで獲物に隣接）を選択しても，その間獲物付近の 2 体のエージェントの停止行動に対する Q 値は減少し続ける．よって，その後のエピソードにおいて部分状態 S に遷移した場合，$\max_a Q(S,a)$ として用いられる Q 値は本来期待される値よりも小さい．Modular Q-learning ではこのような状況が多く存在し，報酬伝播が遅延することにより学習速度が低下する．Modular Q-learning と提案手法の相対的な学習速度は，提案手法の状態数増加による学習速度の低下と，Modular Q-learning の不完全知覚による学習速度の低下 の 2 つの要因により決定される．5x5 グリッド環境では，提案手法の状態数増加の影響が強く反映され，提案手法の学習速度が Modular Q-learning より遅くなった．

一方 9x9 グリッド環境では，表 1 に示すように，Modular Q-learning と提案手法の状態数の差は 5x5 グリッド環境ほど大きなものではなく，さらに提案手

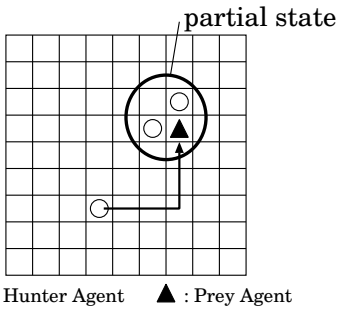


図 8　最適行動の Q 値が低下する状況:獲物は停止していると仮定した場合，丸で囲まれた 2 体のエージェントの最適行動「停止」の Q 値は，左下のエージェントが獲物に隣接するまで減少し続ける．
Fig. 8　The Q-Value of an optimal action decrease.

法では高次元化手法により不完全知覚を取り除くことで，図 8 のような状況においても Q 値の値は減少せず，報酬伝播の遅延による学習速度の低下を防ぐことが可能となったという 2 つの理由により提案手法の学習速度低下が抑えられたと考える．

4. 3. 1　メモリ量と学習性能に関する考察

本手法では高次元状態空間を生成するため，Modular Q-learning よりも多くの状態数を扱う．そのため，学習に必要とするメモリ量も Modular Q-learning に比べて多くなる．表 1 に提案手法，Modular Q-learning，Q-learning が扱う状態数を示す．Modular Q-learning については一つのモジュールが扱う状態数を，提案手法については「一つの 2 次元学習モジュールが扱う状態数＋高次元化された状態数」の値を示す．

表 1　各手法における状態数の比較
Table 1　The number of state

	Modular Q-learning	HMQL	Q-learning
5x5	625	1100	15625
7x7	2401	2920	117649
9x9	6561	7300	531441

表 1 に示すように，Q-learning を用いた場合には非常に多くの状態数を扱わなければいけない．提案手法が扱う状態数は Modular Q-learning に比べてやや多いものの，Q-learning に比べて非常に少ない．このことから，提案手法は十分に少ないメモリ量で実現することが可能であると言える．

表 2 に，各手法における試行終了時の獲物捕獲に要したステップ数を示す．ただし，Q-learning の学習エピソード数は 500 万エピソードとした．いずれの環境においても提案手法のステップ数が最も小さな値となっており，Modular Q-learning の学習性能改善が可能であった．また，本来なら Q-learning のステップ数が最も小さな値となるべきである．この理由として，500 万エピソードでは学習数が不足であったためと考える．表 1，表 2 より，提案手法は十分に少ないメモリ量で優れた学習性能を得ることが可能であった．

表 2　各手法における獲物捕獲に要したステップ数の比較
Table 2　Required steps to capture the prey

	Modular Q-learning	HMQL	Q-learning
5x5	5.60	4.65	4.85
7x7	7.35	6.35	6.45
9x9	9.10	7.90	–

5. ま と め

マルチエージェント環境では，エージェント数が増加するに伴い状態空間が指数的に増大し，学習速度が急激に低下する，膨大なメモリ量が必要となる，といった問題が生じる．Ono らは，この状態空間の爆発を防ぐ手法として Modular Q-learning を提案した．しかし，Modular Q-learning では部分状態のみを観測するため，不完全知覚が生じ学習性能が低下してしまうという欠点がある．

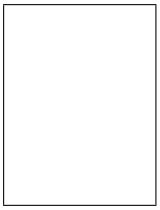
本研究では Modular Q-learning を基盤としたマルチエージェント強化学習法 HMQL を提案した．HMQL では，学習性能低下の要因となる状態を高次元化することで不完全知覚を取り除き，Modular Q-learning の学習性能を改善した．また，追跡問題を用いた実験により，十分な学習速度，少ないメモリ量で性能改善が可能であることを確認した．現段階では 3 次元と比較的低い次元での実験しか行っていないが，$N-1$ 個の 2 次元状態価値 V が閾値を超えた場合に N 次元状態空間を生成することで，より高次元の問題にも適応できると考える．なお，今後の課題として，提案した手法を様々な問題に適用することにより，本手法の有効性を検討する予定である．

文　献

[1] Richard S. Sutton and Andrew G.Barto, “Reinforcement Learning: An Introduction”, A Bradford Book, The MIT Press, 1998
[2] 荒井 幸代, “マルチエージェント強化学習 -実用化に向けての課題・理論・諸技術との融合-”, 人工知能誌, Vol.16, No.4, pp.476–481, 2001
[3] R.H. Crites and A.G. Barto, “Improving elevator performance using reinforcement learning”, Advances

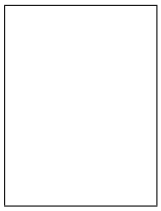
in Neural Information Processing Systems, pp.1017–1023, 1996
[4] D.H. Wolpert, J. Sill, and K. Tumer, "Reinforcement Learning in Distributed Domains: Beyond Team Games," In the Proceedings of the Seventeenth National Conference on Artificial Intelligence, Seattle, WA pp.819–824, 2001
[5] 車谷 浩一, "マルチエージェント社会シミュレーション展望", システム／制御／情報, システム制御情報学会誌, Vol.46, No.9, pp.518–523 2002
[6] 内部 英治, 浅田 稔, 細田 耕, "複数の学習するロボットの存在する環境における協調行動獲得のための状態空間の構成", 日本ロボット学会誌, Vol.20, No.3 pp.281–289, 2002
[7] 枝澤 一寛, 高橋 泰岳, 浅田 稔, "複数学習器を用いたマルチエージェント環境における行動獲得", 第 20 回日本ロボット学会学術講演会予稿集, 3H33, 2002
[8] Ryo Goto, Toshihiro Matsui and Hiroshi Matsuo, "State Generalization with Support Vector Machines in Reinforcement Learning", 4th Asia-Pacific Conference on Simulated Evolution and Learning
[9] 高橋 泰岳, 浅田 稔, "階層型学習機構における状態行動空間の構成", 日本ロボット学会誌, Vol.21, No.2, pp.164–171, 2003
[10] Irfan Gultekin and Ahmet Arslan, "Modular-Fuzzy Cooperation Algorithm for Multi-agent Sustems", Advances in Information Systems: 2nd International Conference, pp.255–263, 2002
[11] 伊藤 昭, 金淵 満, "知覚情報の粗視化によるマルチエージェント強化学習の高速化：ハンターゲームを例に", 信学誌, Vol.J84-D-I No.3, pp285–293, 2001
[12] N. Ono and K. Fukumoto, "Multi-agent Reinforcement Learning: A Modular Approach", in Proceedings of the 2nd International Conference on Multi-agent Systems (ICMAS-96), AAAI Press, 1996
[13] Whitehead, S., Karlsson, J. and Tenenberg, J., "Learning Multiple Goal Behavior via Task Decomposition and Dynamic Policy Merging", Robot Learning, Kluwer Academic Publishers, 1993
[14] 長行 康男, 伊藤 実, "2 体エージェント確率ゲームにおける他エージェントの政策推定を利用した強化学習法", 信学誌, Vol. J86-D-I No.11, 2003

（平成 xx 年 xx 月 xx 日受付）



藤田　和幸

2002 名古屋工業大学電気情報工学科卒. 2004 同大学院博士前期過程了. 在学中, 強化学習, 分散協調システムに関する研究に従事.



松尾　啓志　（正員）

1983 名古屋工業大学情報工学科卒. 1985 同大学院修士課程了. 民間企業を経て 1986 同大学院博士後期課程入学. 1989 同課程了. 1989 名古屋工業大学電気情報工学科助手, 講師, 助教授を経て 2003 同教授, 現在に至る. 分散システム, 分散学習に関する研究に従事

Abstract In Multi-Agent Reinforcement Learning, each agent observe a state of other agents as a part of environment. Therefore, the state space is exponential in the number of agents and learning speed significantly decrease. Modular Q-learning needs very small state space. However, the incomplete observation involves a decline in the performance. In this paper, we improve Modular Q-learning's performance with the partly high-dimensional state space.

Key words Multi Agent, Reinforcement Learning, Modular Q-learning, State Space