Vol.2019-ICS-194 No.3
2019/3/9

マルチエージェント強化学習における全エージェントの経験を用いた状態空間削減手法の検討

岡野拓哉[1,2,a)]　大西正輝[1,2]　野田五十樹[2]

概要：強化学習及びマルチエージェント強化学習は学習に非常に多くの時間を要することが知られている．マルチエージェント強化学習では，複数のエージェントが同時に学習行動を行うため，各エージェントが得た知識を共有，活用することで，学習を高速化することが期待できる．そこで，本研究では，各エージェントから得られた経験を用いて，状態の次元削減を行うオートエンコーダを構築し，そのオートエンコーダを用いて状態空間の次元削減を行うことで，学習速度の高速化を図る手法を提案する．提案手法を Task Allocation Problem 及び，交通信号機制御問題において評価し，通常の Independent Deep Q Learner に比べ学習速度が向上することを確認する．

1. はじめに

近年，強化学習は Atari のゲームで人間以上のスコアを獲得した Deep Q Learning[1] や囲碁で世界チャンピオンに勝利した AlphaGo[2] など様々なタスクにおいて学習可能なことが報告されている．強化学習をパケットルーティング [3] や，交通信号機制御 [4] 等の実問題に適用する際には，制御対象が複数存在するため，マルチエージェント強化学習問題となる．

マルチエージェント強化学習では，複数のエージェントが同時に学習行動を行うため，エージェント数に応じて知識が蓄積される．それらの知識を用いていかに全エージェントの学習性能を向上させるかが重要である．

そこで，本研究では，各エージェントが得た情報を用いて，状態空間を抽象化するオートエンコーダを構成し，全エージェントがそのオートエンコーダを用いることで学習速度を向上させる手法を提案する．最後に提案手法を複数のマルチエージェントゲームを用いて評価する．

2. 準備

2.1 強化学習

強化学習は試行錯誤を行い最適な方策を獲得する学習フレームワークである．強化学習の理論的解析を行うために，マルコフ決定過程 $MDP = \langle S, A, T, R \rangle$ が用いられる．S は状態の集合，A は行動の集合，$R : S \times A \times S \to \mathbb{R}$ は報酬関数，$T : S \times A \times S \to [0, 1]$ は状態遷移関数を表す．強化学習では，現在の状態 $s_t \in S$ において，ある行動 $a_t \in A$ を選択し，実行した後に変化した状態 $s_{t+1} \in S$ 及び報酬 $r_t \in R$ を得ることにより，学習を進めていく．

Q 学習 [5] は遷移先の状態の最大行動価値を用いて学習するアルゴリズムである．Q 学習では行動価値関数 $Q(s_t, a_t)$ を遷移先 $s_{t+1} \in S$ の最大行動価値 $\max_{a'} Q(s_{t+1}, a')$ を用いて次の更新式に従って更新する．

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha(r_{t+1} + \gamma \max_{a'} Q(s_{t+1}, a') - Q(s_t, a_t)).$$

Q 学習は MDP 環境において，最適行動価値に収束することが証明されている [5].

Deep Q Learning(DQL)[1] は深層学習と強化学習を組み合わせた学習アルゴリズムである．DQL では，行動価値関数をニューラルネットワークによって関数近似 $Q(s, a, \theta) \approx Q(s, a)$ して表現する．θ はニューラルネットワークのパラメータを表している．また，Experience Replay を用いて，θ を更新する．Experience Replay では，行動したことによって得られた経験を $\langle s_t, a_t, r_t, s_{t+1} \rangle$ を Experience Buffer D に格納していく．そして，Experience Buffer D から経験をバッチサイズ分取り出し，θ を次式に従って TD 誤差を最小化するように更新していく．

$$\mathcal{L}(\theta) = \frac{1}{|B|} \sum_{e \in B} (r + \gamma \max_{a} Q(s', a; \theta^-) - Q(s, a; \theta))^2,$$

[1] 筑波大学システム情報工学研究科
University of Tsukuba, Tsukuba, Ibaraki 305–0577, Japan
[2] 産業技術総合研究所
National Institute of Advanced Industrial Science and Technology, Tsukuba, Ibaraki 305–8560, Japan
[a)] okano565656@gmail.com

B はサンプリングした経験の集合を表す．

強化学習エージェントはステップ毎にある行動選択手法を用いて行動 a_t を選択する．ϵ-greedy を用いる場合には，ϵ の確率でランダムで行動を選択，$1-\epsilon$ の確率で行動価値が最大化する行動 $a_t = \arg\max_a Q(s, a; \theta^-)$ を選択する．

2.2 マルチエージェント強化学習

マルチエージェント強化学習は複数のエージェントが同時に学習行動を行う学習フレームワークであり，数多く研究されてきた [6]．マルチエージェント強化学習は Markov Games(MG)[7] として定式化される．MG は以下のように定義される．

$$MG = \langle n, S, A_1, ..., A_n, R_1, ..., R_n, T \rangle, \quad (1)$$

n はエージェント数，S は状態の集合，A_i はエージェント i の行動の集合，$R_i : S \times A_1 \times \cdots \times A_n \to \mathbb{R}$ はエージェント i の報酬関数の集合，$T : S \times A_1 \times \cdots \times A_n \to [0, 1]$ は状態遷移関数を表す．各エージェントは，各々の累積報酬の最大化を目指し学習を進める．

マルチエージェント強化学習の最もシンプルな形式は Independent Learner(IL)[8] である．IL では，各エージェントが各々の policy を持ち，他のエージェントを環境の一部として独立で学習する．

マルチエージェント強化学習において DQL を利用する研究も行われている [9], [10]．IL の中でも，全エージェントは DQL によって学習行動を行う場合 Independent Deep Q Learners(IDQL) と呼ぶ．マルチエージェント強化学習では，各エージェントの方策が時間と共に変化するため，過去の経験を学習に用いることができないことがある．そこで，過去の方策と現在の方策によってサンプルの重みを決定する手法 [9] や Experience Buffer のバッファサイズを非常に小さく設定する方法 [10] が提案されている．本研究では，[10] と同様に Experience Buffer のバッファサイズを非常に小さく設定する．

3. 知識の再利用と学習高速化

マルチエージェント強化学習では，複数のエージェントが同時に学習行動を行っているため，各エージェントから得られた知識をいかに利用するかが重要である [11]．

いずれのマルチエージェント強化学習においても，**学習には膨大な時間が必要**であるという問題がある．マルチエージェント強化学習では，各エージェントが得た知識を用いて，学習を高速化することが期待できる．その手法として，エージェント間でアドバイスし合う手法 [12] や，経験を共有する手法 [13] などが提案されてきた．

シングルエージェント強化学習問題においても学習を高速化する仕組みとして，状態空間を削減する手法を提案している [14]．文献 [14] では，オートエンコーダを用いて次元削減を行う手法を提案している．しかしながら，オートエンコーダを学習するために膨大なサンプルが必要であるため，オートエンコーダの学習に時間がかかる．

そこで，本研究では，全エージェント得た経験を用いて，状態の次元削減を行うオートエンコーダを学習することで，オートエンコーダ及び強化学習エージェントの学習を高速化する手法を提案する．

4. 提案手法

全エージェントの経験を用いて，状態空間の次元削減を行うオートエンコーダを構成し学習を行う手法を提案する．全エージェントの経験を用いることで，状態空間の重なりが多ければ多いほど，効率よくオートエンコーダを学習することができる．つまり，類似した環境で学習行動を行うエージェントが多いほどオートエンコーダの学習を高速化できる．

提案手法の構成図を図 1 に示す．状態を圧縮するオートエンコーダを全エージェントで共有している．全エージェントの経験を用いてオートエンコーダを学習し，このオートエンコーダを用いて，全エージェントは状態空間の次元削減を行う．このエンコーダを Shared State Encoder(SSE) と呼ぶ．

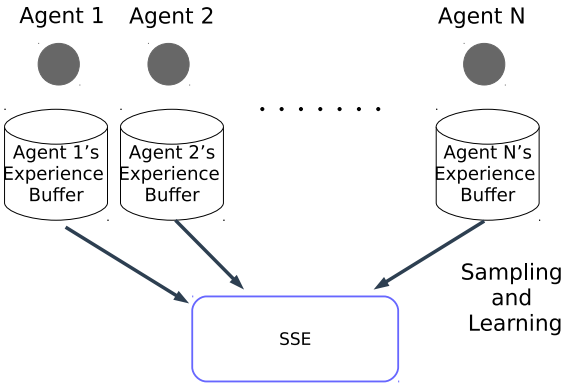

図 1 提案手法の構成

4.1 Shared State Encoder

SSE は各エージェントが得た状態を入力し，その状態を再現するように学習する．そして，全エージェントは SSE に自由にアクセス可能であると仮定する．SSE を高速で学習するために，全エージェントが得た経験を用いて，SSE を学習する．つまり，SSE を学習する際には，全エージェントの Experience Buffer から経験をサンプリングし，サンプリングした経験の中の状態 s を用いて学習する．全エージェントの経験を用いるため，状態空間の重なりが多ければ多いほど，効率よくオートエンコーダを学習できる．

4.2 SSE を用いた IDQL

SSE を用いた IDQL では, 各エージェントは通常の IDQL と同様に現在の状態 s_t を観測し，状態 s_t を SSE を用いて次のように変換する．

$$z_t \leftarrow SSE(s_t; \theta_t^{SSE}), \tag{2}$$

ここで，θ_t^{SSE} は t ステップ時の SSE のパラメータである．

また，Experience Replay を行う際には，エージェントが Experience Buffer からバッチサイズ分経験をサンプリングし，すべての経験の状態を SSE によって変換する．変換した状態を用いて DQL と同様に Q-Network を学習する．

提案手法を用いた IDQL(IDQL+SSE) のアルゴリズムを Algorithm 1 に示す．

Algorithm 1 IDQL using SSE

for step $t = 1$ to T **do**
  Each agent i converts current state s_{t-1}^i to z_{t-1}^i using Eq(2)
  Each agent i chooses action a_t^i using $Q^i(z_{t-1}^i; \theta_t^-)$
  Execute actions $a_t = (a_t^1, ..., a_t^n)$
  Each agent i observe s_t^i, r_t^i and stores $\langle s_{t-1}^i, a_{t-1}^i, r_t^i, s_t^i \rangle$ to D^i
  for agent $i = 0$ to N **do**
    Sample B experiences $\langle s, a, r, s' \rangle$ from D^i
    Convert B experiences $\langle s, a, r, s' \rangle$ to $\langle z, a, r, z' \rangle$ using Eq(2)
    Update Q-network by minimizing the loss $\mathcal{L}(\theta_t) = \frac{1}{|B|} \sum_{e \in B} (r + \gamma \max_{a'} Q(z', a'; \theta_t^-) - Q(z, a; \theta_t))^2$
    Update the parameters of the target network in the interval I
  end for
  Update SSE's parameter θ_t^{SSE} using all agent's experience
end for

5. 評価用問題

提案手法を評価するために Task Allocation Problem 及び交通信号機制御問題を用いる．

5.1 Task Allocation Problem

Task Allocation Problem(TAP) はグリッドコンピューティングやネットワークルーティングなどを抽象化したマルチエージェントゲームの 1 つである．TAP を以下のように定義する．

$$TAP = < \mathcal{T}, \mathcal{R}, \mathcal{C}, \mathcal{S}, \mathcal{A} >, \tag{3}$$

$\mathcal{T}$ はタスクの集合であり，各タスク $\mathcal{T}_i \in \mathcal{T}$ は $\mathcal{T}_i = \langle t_0^{\mathcal{T}_i}, ..., t_k^{\mathcal{T}_i} \rangle$ ようにベクトルで表現する．$\mathcal{R} = \{\mathcal{R}_0, ..., \mathcal{R}_m\}$ は資源の集合を表している．$\mathcal{C}$ は資源のキャパシティであり，資源 $\mathcal{R}_i \in \mathcal{R}$ のキャパシティは $\mathcal{C}_{\mathcal{R}_i} = \langle c_0^{\mathcal{R}_i}, ..., c_k^{\mathcal{R}_i} \rangle$ となり，タスクの各属性に対応する値を保持している．$\mathcal{S}$ はスケジューラの集合であり，各資源に配置されており，環境から得たタスクをどの資源に割り振るかを決定する．$\mathcal{A} = \{\mathcal{A}_{ij}\} \in \mathbb{R}^{m \times m}$ はスケジューラ間の隣接関係を表している．各資源はタスクの実行待ちキュー $\mathcal{Q}_{\mathcal{R}_i}$ を保持している．

各スケジューラは，ステップ毎に環境からある確率に従ってタスクを得る．そのタスクを各スケジューラに対応している資源，もしくは，隣接している資源の実行待ちキューに挿入する．各資源はステップ毎に待ちキューからタスクを取り出し実行していく．資 源 $\mathcal{R}_i$ の待ちキュー $\mathcal{Q}_{\mathcal{R}_i}$ にあるすべてのタスクの実行が完了するまでの時間を Load of Resource(LoR) と呼び，次式で定義する．

$$LoR_i = \frac{1}{|\mathcal{Q}_{\mathcal{R}_i}|} \sum_{\mathcal{T}_j \in \mathcal{Q}_{\mathcal{R}_i}} \max(t_0^{\mathcal{T}_j}/c_0^{\mathcal{R}_i}, ..., t_k^{\mathcal{T}_j}/c_k^{\mathcal{R}_i}). \tag{4}$$

この問題では全資源に割り振られた全タスクの期待終了時間 Average Load of Resources($ALoR$)[15] という全タスクの期待終了時間をシステムの性能指標として用いる．$ALoR$ は以下のように定義される．

$$\begin{aligned} ALoR &= \frac{1}{|\mathcal{R}|} \sum_{\mathcal{R}_i \in \mathcal{R}} LoR_i \\ &= \frac{1}{|\mathcal{R}|} \sum_{\mathcal{R}_i \in \mathcal{R}} \frac{1}{|\mathcal{Q}_{\mathcal{R}_i}|} \sum_{\mathcal{T}_j \in \mathcal{Q}_{\mathcal{R}_i}} \max(t_0^{\mathcal{T}_j}/c_0^{\mathcal{R}_i}, ..., t_k^{\mathcal{T}_j}/c_k^{\mathcal{R}_i}). \end{aligned}$$

この問題では $ALoR$ を最小化するように各スケジューラがタスクを適切に配分することを目指すゲームである．

TAP における強化学習エージェントの設定について述べる．強化学習エージェントはスケジューラを適切に制御して，タスクを効率よく処理することが目標となる．強化学習エージェントの行動空間，状態空間及び報酬関数を定義する．

行動空間: エージェントの行動空間はエージェントが制御しているスケジューラに対応している資源及び隣接している資源とする．つまり，各エージェントは環境から得られたタスクを制御しているスケジューラに対応した資源，もしくは隣接している資源に割り振るかを決定する．

状態空間: エージェントの状態は，エージェントが制御するスケジューラ及び隣接したスケジューラに対応する資源に割り振られたタスクと資源の LoR の集合とする．

報酬: 各エージェント i の報酬は，エージェントが選択した資源 j の LoR_j を用いて以下のように計算する．

$$R_i = 1 - LoR_j. \tag{5}$$

つまり，エージェントはタスクが最も早く処理される資源を選択したときに最も高い報酬を得られる．ここでの LoR は各スケジューラに隣接した資源の LoR を用いて正規化した値とする．

5.2 交通信号機制御問題

交通信号機制御問題は信号機のスプリットを適切に制御することで車両の待ち時間を減らすことを目指す問題である．

信号機には複数の制御対象のパラメータが存在する．大きく分けると，青から赤までの一連の流れの長さを決定するサイクル長，サイクル内の各現示の比率を決定するスプリット，隣接した信号機間の青信号の開始時間のずれを決定するオフセットの 3 つの制御対象のパラメータが存在する．本研究では，スプリットを制御することで，交通渋滞を減少させることを目指す．つまり，サイクル内の各現示の比率を調整することで，交通渋滞を減少させることを目指す．

この問題では，各エージェントが各信号機を制御する．各エージェントは，各サイクルが始まる前に次のサイクルのスプリットをコントロールする．

信号機を強化学習によって制御するために，行動空間，状態空間，報酬について定義する．

行動空間: 南北の青の比率を NS，東西の青の比率を EW として，次の 3 つのスプリットを行動空間とする．

(1) $NS = EW$
(2) $NS > EW$
(3) $NS < EW$

各エージェントはサイクル毎に，この 3 つの中から 1 つを選択する．

状態空間: 状態は，各エージェントが制御している信号機の 1 サイクルの交差点の流入方向のレーン (流入レーン) の平均車両占有率と平均停車数により表現する．

報酬: 各エージェントが制御している信号機がある交差点の流入レーンの 1 サイクル Δt の平均車両待ち時間を元に，以下のようにエージェント i の報酬 R_i を定義する．

$$R_i = -\frac{1}{\Delta t}\sum_{k=t}^{t+\Delta t}\frac{1}{|L_i|}\sum_{l\in L_i} w_k^l \ , \tag{6}$$

L_i は信号機 (エージェント) i が制御している流入レーンの集合，w_t^l は t ステップ時のレーン $l \in L_i$ を走行している車両の待ち時間を表している．各エージェントがコントロールしているレーン上にいる車両の待ち時間が少なければ少ないほど高い報酬を与える．

6. 実験と考察

提案手法を TAP 及び，交通信号機制御問題において評価する．比較手法として，通常の IDQL を用いる．Q-Network の隠れ層を 2 層として，各層の素子数は 16-16 とする．提案手法を用いた IDQL(IDQL+SSE) は入力状態の次元圧縮を行っているため，各層の素子数は 4-4 とする．強化学習エージェントのパラメータを表 1 に示す．

表 1 強化学習のパラメータ
Table 1 Parameter of learning agents

Parameter	Value
Initial exploration rate (ϵ)	1.0
ϵ decay rate	0.99
Lowest ϵ	0.001
Discount factor (γ)	0.99
Optimizer	Adam
Learning rate	0.001
Interval of updating target network(I)	16
Size of experience buffer	1000
Batch size	32

6.1 Task Allocation Problem

本実験では，二種類のキャパシティ $C = \{\langle 0.8, 0.6, 0.4, 0.2\rangle, \langle 0.2, 0.4, 0.6, 0.8\rangle\}$ のどちらかを持つ資源を用意し，格子状のネットワークにランダムで配置されている TAP を用いる．タスクはポアソン分布に従って生成され，生成されるタスクの各要素の値は区間 $(0, 1)$ の一様乱数によって決定する．本実験では，2 つの格子状のネットワークを用いて評価を行う．図 2 に本実験で用いる 4×4 及び 5×5 のネットワークを示す．

本実験では，1 エピソードは 10000 ステップとし，10 エピソード学習行動を行う．複数回同じ設定を用いて実験を行い評価する．

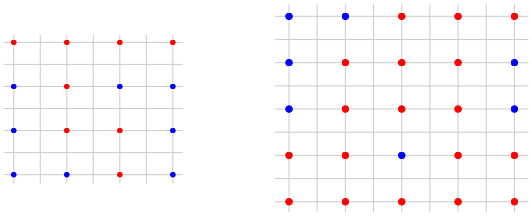

図 2 左が 4×4 グリッドネットワーク，右が 5×5 グリッドネットワーク．各丸が資源を表す．赤い資源のキャパシティは $C^{red} = \langle 0.8, 0.6, 0.4, 0.2\rangle$，青い資源のキャパシティは $C^{blue} = \langle 0.2, 0.4, 0.6, 0.8\rangle$

Fig. 2 Left is 4×4 grid network, Right is 5×5 grid network. Each circle represents a resource. The capacity of red resource is $C^{red} = \langle 0.8, 0.6, 0.4, 0.2\rangle$. The capacity of red resource is $C^{blue} = \langle 0.2, 0.4, 0.6, 0.8\rangle$

4×4 及び 5×5 のネットワークによる実験結果を図 3 に示す．横軸はエピソード，縦軸はエピソード毎の全資源の平均 $ALoR$ を表している．太線が平均 $ALoR$，薄い領域が標準偏差を表している．青が通常の IDQL，赤が提案手法を用いた IDQL(IDQL+SSE) を表す．

両手法ともに $ALoR$ を減少させることができている．提案手法を用いた IDQL(IDQL+SSE) は，通常の IDQL に比べて，少ない時間で Average $ALoR$ を小さくしていること

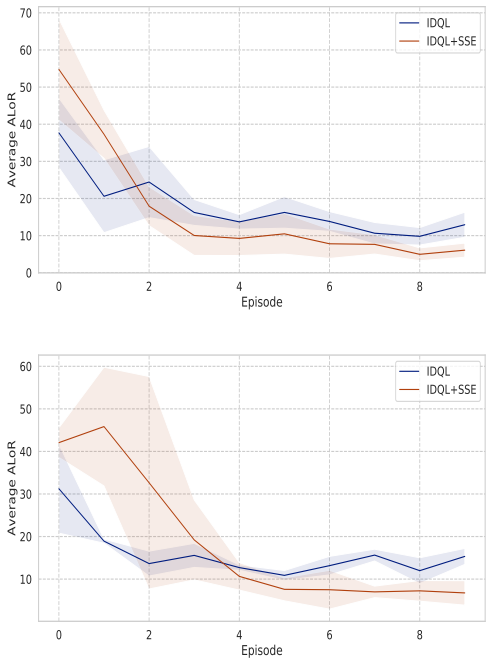

図 3 エピソード毎の *ALoR* の推移．上が 4×4 グリッドネットワーク，下が 5×5 グリッドネットワークによる実験結果

Fig. 3 Change of *ALoR* in each episodes. Upper graph is the result in 4×4 grid network, Lower is the result in 5×5 grid network

がわかる．しかしながら，序盤のエピソードにおける性能は乏しい．これは，状態を抽象化する SSE の学習が十分でなく，正しく状態をエンコードできていないことが原因であると考えられる．

6.2 交通信号機制御問題

先に述べた，マルチエージェント強化学習による信号機制御を交通シミュレーター SUMO(Simulation of Urban MObility)[16] を用いて評価する．

本研究では 3×3 のシンプルな格子状のマップを用いる．実験で用いるマップを図 4 に示す．各レーンの長さは 100m，制限速度は 40km/h，1 サイクルを 90 秒とする．表 2 に本実験で用いる OD 表を示す．1 エピソードをすべての車両が生成されてから，目的地に到達するまでとし，10 エピソード試行する．複数回同じ設定で実験を行い評価する．

表 2 実験で用いる OD 表

Table 2 OD matrix which is used in the experiments

O \ D	A	B	C	D	E	F	G	H	I	J	K	L
A	0	5	1	4	3	4	3	5	200	1	3	5
B	4	0	1	3	4	5	2	100	2	1	3	5
C	2	2	0	5	1	1	50	2	4	4	5	4
D	1	3	3	0	3	5	4	3	2	5	1	200
E	3	1	1	2	0	2	3	4	1	4	100	5
F	5	4	3	5	1	0	3	5	4	50	5	2
G	1	1	50	5	4	2	0	2	4	3	2	4
H	4	100	2	5	3	4	5	0	1	2	1	5
I	200	3	4	2	1	2	4	3	0	2	2	5
J	1	5	3	4	1	50	2	2	3	0	3	2
K	3	3	1	5	100	5	2	1	5	5	0	5
L	4	1	4	200	5	1	3	4	5	5	4	0

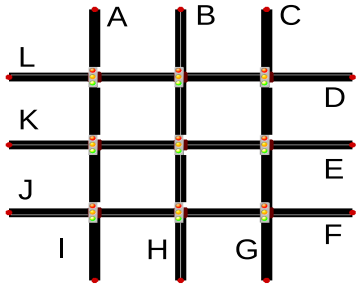

図 4 実験で用いるマップ．アルファベットは車両の出発地点もしくは到着地点を表す

Fig. 4 Map is used to evalute the proposal method. Alphabet represents origin-destination of cars

実験結果を図 5 に示す．横軸がエピソード，縦軸はエピソード毎の平均車両時間を表す．太線が平均車両待ち時間，薄い領域が標準偏差を表している．青が通常の IDQL，赤が提案手法を用いた IDQL(IDQL+SSE) を表す．また，後半の平均待ち時間を表 3 に示す．図 5 から，両手法ともに平均車両待ち時間を減少させることに成功している．僅かではあるが，IDQL+SSE によって信号機を制御することで，車両の平均待ち時間を IDQL に比べて少なくできている．TAP における実験結果と同様に IDQL に比べ，序盤の性能は乏しい．しかしながら，表 3 からエピソード後半においては，IDQL に比べて良い性能が得られていることがわかる．本実験においては，エージェント数が 9 体であり，さらに類似した環境において学習行動をしているエージェント数が少ないため，オートエンコーダの学習に時間が要していることが考えられる．

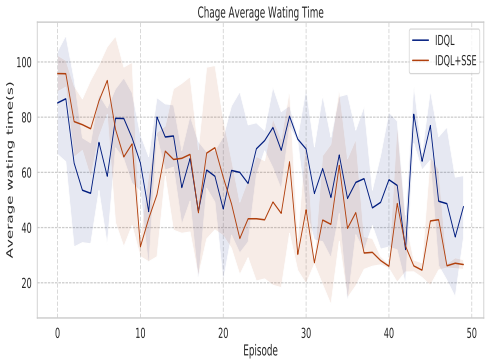

図 5 エピソード毎の平均車両待ち時間の推移

Fig. 5 Change of the average waiting time of cars in each episode

表 3 エピソード後半の平均待ち時間

Table 3 Average waiting time of cars in each episodes

Name	Average ALoR(Standard Error)
IDQL	59.10(16.90)
IDQL+SSE	38.01(14.87)

6.3 考察

提案手法によって IDQL に比べ学習を高速化可能なことを実験的に示した．

しかしながら，本稿で紹介した手法は，オートエンコーダによってエージェントが学習すべき状態空間が時間経過とともに変化していくという問題点がある．つまり，エージェントが学習している状態空間が変化していくため，学習自体が不安定になる．そのため，TAP 及び交通信号機制御問題の両問題において，序盤の性能が著しく低かったことが考えられる．

また，本手法を適用する問題設定では，各エージェントの状態空間が類似している必要がある．各エージェントの状態空間が非常に異なる場合，オートエンコーダの学習に非常に時間がかかるため，エージェントの学習速度の高速化は期待できない．そのため，各エージェントの環境が類似している問題を解くエージェントが多いマルチエージェント強化学習に適用するべきである．

7. おわりに

本稿では，マルチエージェント強化学習において，全エージェントから得られた経験を用いて，状態を抽象化するエンコーダを構成し，そのエンコーダを用いて学習を高速化する手法を提案した．そして，提案手法を Task Allocation Problem 及び，交通信号機制御問題 を用いて評価を行った．結果として，提案手法を用いた IDQL は，通常の IDQL に比べ，学習速度を向上させることを確認した．しかしながら，本稿では，シンプルな設定のみにおいての評価であったため，今後はより実問題に近い設定においての評価を行う必要がある．

謝辞 この成果は，国立研究開発法人新エネルギー・産業技術総合開発機構 (NEDO) の委託業務の結果得られたものです．

参考文献

[1] Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., Petersen, S., Beattie, C., Sadik, A., Antonoglou, I., King, H., Kumaran, D., Wierstra, D., Legg, S. and Hassabis, D.: Human-level control through deep reinforcement learning, *Nature*, Vol. 518, No. 7540, pp. 529–533 (2015).

[2] Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., van den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., Dieleman, S., Grewe, D., Nham, J., Kalchbrenner, N., Sutskever, I., Lillicrap, T., Leach, M., Kavukcuoglu, K., Graepel, T. and Hassabis, D.: Mastering the game of Go with deep neural networks and tree search, *Nature*, Vol. 529, pp. 484–503 (2016).

[3] Boyan, J. A. and Littman, M. L.: Packet Routing in Dynamically Changing Networks: A Reinforcement Learning Approach, *Proceedings of the 6th International Conference on Neural Information Processing Systems*, NIPS'93, San Francisco, CA, USA, Morgan Kaufmann Publishers Inc., pp. 671–678 (1993).

[4] Van der Pol, E. and Oliehoek, F. A.: Coordinated Deep Reinforcement Learners for Traffic Light Control, *NIPS'16 Workshop on Learning, Inference and Control of Multi-Agent Systems* (2016).

[5] j.C.H, C. and Dayan, P.: Technical Note: Q-Learning, *Machine learning*, Vol. 8, No. 3-4, pp. 279–292 (1992).

[6] Busoniu, L., Babuska, R. and Schutter, B. D.: A Comprehensive Survey of Multiagent Reinforcement Learning, *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, Vol. 38, No. 2, pp. 156–172 (2008).

[7] Shapley, L. S.: Stochastic Games, *Proceedings of the National Academy of Sciences*, Vol. 39, No. 10, pp. 1095–1100 (1953).

[8] Tan, M.: Readings in Agents, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, chapter Multi-agent Reinforcement Learning: Independent vs. Cooperative Agents (1998).

[9] Foerster, J., Nardelli, N., Farquhar, G., Afouras, T., Torr, P. H. S., Kohli, P. and Whiteson, S.: Stabilising Experience Replay for Deep Multi-Agent Reinforcement Learning, *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, pp. 1146–1155 (2017).

[10] Leibo, J. Z., Zambaldi, V., Lanctot, M., Marecki, J. and Graepel, T.: Multi-agent Reinforcement Learning in Sequential Social Dilemmas, *Proceedings of the 16th Conference on Autonomous Agents and MultiAgent Systems*, AAMAS '17, Richland, SC, International Foundation for Autonomous Agents and Multiagent Systems, pp. 464–473 (2017).

[11] Silva, F. L. D., Taylor, M. E. and Costa, A. H. R.: Autonomously Reusing Knowledge in Multiagent Reinforcement Learning, *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence*, IJCAI '18, International Joint Conferences on Artificial Intelligence Organization, pp. 5487–5493 (2018).

[12] da Silva, F. L., Glatt, R. and Costa, A. H. R.: Simultaneously Learning and Advising in Multiagent Reinforcement Learning, *Proceedings of the 16th Conference on Autonomous Agents and MultiAgent Systems*, AAMAS '17, International Foundation for Autonomous Agents and Multiagent Systems, pp. 1100–1108 (2017).

[13] Garant, D., da Silva, B. C., Lesser, V. and Zhang, C.: Context-Based Concurrent Experience Sharing in Multiagent Systems, *AAMAS Proceedings of the 16th Conference on Autonomous Agents and MultiAgent Systems*, AAMAS '17, pp. 1544–1546 (2017).

[14] Lange, S. and Riedmiller, M. A.: Deep auto-encoder neural networks in reinforcement learning., *IJCNN*, IEEE, pp. 1–8 (2010).

[15] Wu, J., Xu, X., Zhang, P. and Liu, C.: A Novel Multi-agent Reinforcement Learning Approach for Job Scheduling in Grid Computing, *Future Gener. Comput. Syst.*, Vol. 27, No. 5, pp. 430–439 (2011).

[16] Behrisch, M., Bieker, L., Erdmann, J. and Krajzewicz, D.: SUMO - Simulation of Urban MObility: An overview, *in SIMUL 2011, The Third International Conference on Advances in System Simulation*, pp. 63–68 (2011).