補助的なエージェントを用いたマルチエージェント強化学習

中田 惇貴[1,a)]　鶴岡 慶雅[1]

概要：複数のエージェントを同時に学習させる強化学習手法であるマルチエージェント強化学習は現実世界への応用を考える際に非常に重要であり, 近年研究が盛んに行われている. マルチエージェント強化学習に用いられるアルゴリズムはしばしば複数のエージェントを特に区別せずに学習を行うことが多い. その結果, 複数のエージェントが同じような行動をとってしまい, 一定の報酬は獲得できるものの, 報酬を最大化するような方策を学習できず局所的最適解に陥ってしまう場合がある. 本研究においては, 各エージェントの報酬の共有のされ方に着目し, 各エージェントを区別して扱うために, 自身の報酬よりむしろ他者の報酬を獲得することを目指す補助的なエージェントを導入することを提案する. 実験環境としては, 協調的なマルチエージェント強化学習の環境である PressurePlate を選択し, 提案手法の効果の検証を行った. その結果として, 提案手法により既存手法と比較して性能が向上することが確認された.

Multi-Agent Reinforcement Learning with Assistive Agents

Atsuki Nakata[1,a)]　Yoshimasa Tsuruoka[1]

Abstract: Multi-agent reinforcement learning, which is a reinforcement learning method to train multiple agents simultaneously, is very important for real-world applications and has been studied extensively in recent years. Algorithms used in multi-agent reinforcement learning often train multiple agents without distinguishing between them. As a result, the multiple agents may behave in the same way, and although a certain reward can be obtained, it may not be possible to learn a strategy to maximize the reward, resulting in a locally optimal solution. In this study, we focus on how the rewards are shared by each agent and propose to introduce an assistive agent that aims to obtain others' rewards rather than its own rewards, in order to treat each agent differently. As an experimental environment, we selected PressurePlate, an environment for cooperative multi-agent reinforcement learning, to verify the effectiveness of the proposed method. As a result, it is confirmed that the proposed method improves the performance compared with the existing methods.

1. はじめに

近年, 強化学習の分野において深層学習を導入することにより, 様々なタスクを対象とする実験において, 性能が大きく改善された [1], [2]. 強化学習を適応する対象としては, Atari2600 といったゲームや, 囲碁, ロボットの制御のようなものが挙げられる. 性能に関しては, 例えば Atari2600 においては近年, 57 個すべてのゲームにおいて人間を上回るようなスコアを達成するような方策を学習することができることが主張されている [3].

強化学習は与えられた環境に対して行動を繰り返し行うことにより, 環境から得られた情報から, どのような行動を選択すべきかという行動決定を行う方策の改善を行う手法である. 強化学習はゲームやロボット制御 [4] といったタスクにおいて, 優れた性能を発揮する手法である一方で, 一般の強化学習では単一のエージェントの方策を改善することを目標に学習が行われるため, 現実世界への応用といった点において課題がある.

それに対して, 複数のエージェントを同時に学習させ, 環境から得られる報酬を最大化しようという試みであるマルチエージェント強化学習の研究が盛んに行われている

[1] 東京大学大学院情報理工学系研究科電子情報学専攻
Department of Information and Communication Engineering, Graduate School of Information Science and Technology, The University of Tokyo
[a)] nakata@logos.t.u-tokyo.ac.jp

©2022 Information Processing Society of Japan

[5], [6]. マルチエージェント強化学習では, 他のエージェントとの協力を行う協調的な環境 [7] と, 他のエージェントと敵対関係にある状況で自身が得られる報酬を最大化するような競争的な環境 [8] の 2 つの環境を対象として研究が行われている. 本稿で協調的な環境での学習を想定する.

マルチエージェント強化学習は現実世界への応用や StarCraft II といったゲームへの応用 [9] という点において非常に重要である一方, 一般に各エージェントを特に区別せずに学習を行うことが多く, こういった場合, 一定以上の報酬を獲得した場合, さらなる探索が促進されず, 報酬を最大化することが難しくなるといった課題がある [10]. また, より複雑な環境では各エージェントが同じような行動をとってしまうだけでは, 報酬があまり得られないといった場合も考えられる.

このような課題を踏まえ, 本稿では, マルチエージェント強化学習の枠組みにおいて, エージェントを全て同じものとして扱うのではなく, エージェントに明示的に役割を与える手法について検討を行った. 具体的には, 自身の報酬獲得を第一に考えるように学習を行う一般のエージェントの他に, 自身の報酬よりもむしろ他者の報酬を獲得することを援助する補助的なエージェントを導入し, 単純な環境においてどのような学習が行われるのかについての実験を行った.

2. 背景

2.1 強化学習

強化学習とは, 機械学習の一分野であり, ある環境に対して, 得られる累積報酬が最大となるような行動系列を獲得することを目的としている.

機械学習は大きく分けて, 教師あり学習, 教師なし学習, 強化学習の 3 つに分類される. 教師あり学習と, 教師なし学習との違いという観点で強化学習について説明する. まず教師あり学習は, データ集合とそれらに対する正解ラベルが与えられ, そのデータをもとに学習を行うことにより, 未知のデータがどのラベルに属するデータであるのかについて分類することが目的となる. また, 教師なし学習では, 正解ラベルが与えられず, データ集合のみが与えられる.

一方, 強化学習では, 一般的には事前にデータが与えられず, 実際に環境に対して試行錯誤を行うことによって, データや報酬を獲得し, それらを用いて, 最適な行動を出力する方策を学習することを目的とする. このように, 主体的にデータを集めるフェーズが存在するという点において, 教師あり学習や教師なし学習と区別することが可能である.

2.1.1 マルコフ決定過程

強化学習における枠組みとして一般的にマルコフ決定過程 (Markov Decision Process, MDP) が用いられる. これは, ある状態は 1 つ前の状態と行動によってのみ決定されるというものである. 強化学習における MDP は, 状態集合である $\mathcal{S}$, 行動の集合である $\mathcal{A}$, ある状態 $s \in \mathcal{S}$, 行動 $a \in \mathcal{A}$ に対して, 次にどのような状態 $s' \in \mathcal{S}$ に遷移するのかという遷移関数 $T(s', s, a)$, 同じく行動, 状態に対して, どのような報酬が与えられるのかという報酬関数 $R(s, a)$ によって定められる.

2.1.2 累積報酬和

強化学習では, 環境から与えられた状態をもとに, 環境に対してエージェントが何かしらの行動を実行し, その結果エージェントは環境から報酬と次の状態が与えられる. これらの一連の流れを 1 ステップとし, 環境の状態が終端状態になるか, 一定のステップ数が経過するまで繰り返す. このように環境の開始から終了までをまとめてエピソードと言う. また, あるエピソードの中で, 時刻 t における状態を s_t, 行動を a_t, 報酬を r_t とそれぞれ表記する. ある状態 s が与えられたとき, エージェントがどのような行動 a を選択するかという方策を確率分布 $\pi(a \mid s)$ と表す. さらにエピソードが時刻 T で終了する場合, ある時刻 t における報酬の総和を, 割引率と呼ばれる, 0 以上 1 未満である γ を用いて, 以下のように表記する.

$$
\begin{aligned}
G_t &= r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + ... + \gamma^{T-t-1} r_T \\
&= \sum_{k=t+1}^{T} \gamma^{k-t-1} r_k
\end{aligned}
$$

強化学習の枠組みにおいては, この累積報酬和を最大化することを目的として学習を行う.

2.2 マルチエージェント強化学習

一般の強化学習では, エージェントは単一のものを想定される場合が多い. 一方で複数のエージェントを同時に学習させる学習手法も存在する. このような手法をまとめて, マルチエージェント強化学習と呼ぶ. エージェントが複数存在する環境は現実世界への応用を考える際に非常に重要であり, 近年研究が盛んに行われている.

マルチエージェント強化学習が対象とする環境は, 大きく分けて 2 つの環境に区別することが可能である. 一方の環境は他のエージェントとの協力を行うことによって高い報酬を得るような方策を学習する協調的な環境であり [11], 他方の環境は, 他のエージェントと敵対関係にある状況で自身が得られる報酬を最大化するような競争的な環境である [12]. 本稿では協調的な環境において行われた研究について紹介する.

実世界への応用を考える上で, マルチエージェント強化学習は非常に重要である一方で, 学習には複数のエージェントそれぞれについての方策を学習させる必要があるため, 計算量などの関係で一般に単一エージェントを扱う場合と比較して難しいことが知られている.

2.2.1 マルコフゲーム

マルコフ決定過程の枠組みは, 単一のエージェントを対

©2022 Information Processing Society of Japan

象とした強化学習に対して適用される枠組みである．この枠組みをマルチエージェントに対して適応できる形に拡張したものも存在する．これはマルコフゲームと呼ばれるものである [13].

状態の集合である $\mathcal{S}$, 各エージェントごとに行動の集合である $\mathcal{A}_i$ $(i = 1, 2, \ldots n)$, ある状態 $s \in \mathcal{S}$, 行動の集合 $a_i \in \mathcal{A}_i$ $(i = 1, 2, \ldots n)$ に対して, 次にどのような状態 $s' \in \mathcal{S}$ に遷移するのかという遷移関数 $T(s', s, a_1, a_2, \ldots, a_n)$, 同じく, 状態, 行動の集合に対して, どのような報酬が与えられるのかという報酬関数 $R(s, a_1, a_2, \ldots, a_n)$ によって定められる.

2.2.2 累積報酬和

マルチエージェント強化学習においても, 単一のエージェントを対象とする一般の強化学習と同様, 2.1.2 項で紹介したように, 累積報酬和を最大化することによって, 学習を行う.

その際, 2.1.2 項では, MDP に従い, 環境から得られる報酬を $R(s, a)$ とすることによって報酬を計算するが, マルチエージェント強化学習の枠組みでは, マルコフゲームに従い, $R(s, a_1, a_2, \ldots, a_n)$ を用いる.

3. 関連研究

3.1 Multi-Agent Deep Deterministic Policy Gradient (MADDPG)

マルチエージェント強化学習のアルゴリズムとして有名なものとして, Multi-Agent Deep Deterministic Policy Gradient (MADDPG) [14] がある．単一のエージェントに対して用いられるアルゴリズムである Deterministic Policy Gradient (DDPG) [15] を, 複数のエージェントに対して適応可能な形にした手法である．このアルゴリズムでは, Actor-Critic [16] のフレームワークに従い動作する．また, 学習時にあるエージェントが他の全てのエージェントの観測と行動が既知の情報として与えられることを前提としている.

Critic と Actor はそれぞれエージェントの数だけ存在し, あるエージェントに対応する Critic の学習には, 他の全てのエージェントの観測, 行動をデータとして学習に用いる．各 Actor に関しては, 学習された Critic の情報を用いることで更新を行う．また, 実際に行動をする際には学習された Actor に従い, 自身の観測のみから行動を選択する．また, 今回の設定のように, Critic が対応するエージェントだけでなく, 他のエージェントの観測, 行動の履歴を利用するモデルを中央監視モデルと呼ぶ.

この論文が扱ったタスクとしては, 障害物が存在する状況下で, 速く動く一つの目標を, ゆっくり動く複数のエージェントが協調的に狩るといったタスクや, エージェントと同数の目標が存在し, それらの目標をエージェントが占める, つまり各エージェントがそれぞれ別の目標の近くに行くことが求められる環境といったタスクなどがある．前者のタスクにおいては, 単一のエージェントだけでは速さの関係上, 獲物を狩ることができないことから, 後者のタスクでは, それぞれのエージェントが勝手に動いてしまうと全ての目標を占めることができないことからどちらのタスクもマルチエージェント強化学習のタスクとして適していると言える.

MADDPG ではこういったタスクにおいて, 単一のエージェントをそれぞれ個別に学習させ, 複数のエージェントが存在する環境に適応させた場合と比較して高い性能を示すことが実験により確認された.

MADDPG において最大化する目的関数は, すべてのエージェントの観測を $\mathbf{x}$, μ_i を各エージェントの方策, μ をエージェントの方策全体, 各エージェントの方策を構成するパラメータを $\boldsymbol{\theta}_i$, 各エージェントの方策をそれぞれ π_i, 状態の分布を p^μ とすると, 以下のように記述することができる.

$$J(\boldsymbol{\theta}_i) = \mathbb{E}_{s \sim p^\mu, a_i \sim \pi_i}[Q_i(\mathbf{x}, a_1, \ldots a_N)]$$

両辺をを $\boldsymbol{\theta}_i$ で微分すると, 更新の際に用いる勾配は以下の式で与えられる.

$$\nabla_{\boldsymbol{\theta}_i} J(\boldsymbol{\theta}_i) = \mathbb{E}_{s \sim p^\mu, a_i \sim \pi_i}[\nabla_{\boldsymbol{\theta}_i} \log \pi_i(a_i|o_i) Q_i(\mathbf{x}, a_1, \ldots a_N)]$$

MADDPG は決定的な方策をとる off-policy な手法であるため, リプレイバッファを $\mathbb{D}$, として与式を書き換えると以下の式を得る.

$$\nabla_{\boldsymbol{\theta}_i} J(\boldsymbol{\theta}_i) = \mathbb{E}_{\mathbf{x}, a_i \sim \mathbb{D}}[\nabla_{\boldsymbol{\theta}_i} \mu_i(a_i|o_i) \nabla_{\boldsymbol{a}_i} Q_i^\mu(\mathbf{x}, a_1, \ldots a_N)|_{a_i = \mu_i(o)}]$$

また, Critic の更新は以下の TD 誤差を最小化することにより行われる.

$$\mathcal{L}(\theta_i) = \mathbb{E}_{x,a,r,x'}[(Q^{\mu_\mathrm{i}}(\mathbf{x}, a_1, \ldots a_N) - y^2)]$$

なお, x, a, r, x' はリプレイバッファからサンプルした値である．また, このとき y はターゲット方策 μ' を用いて以下のように与えられる.

$$y = r_i + \gamma Q_i^{\mu'}(\mathbf{x}', a_1', \ldots a_i', \ldots, a_N')|_{a' = \mu'(o)}$$

本稿で行った実験は, MADDPG における手法をベースラインとして行われた.

3.2 エージェントの行動の多様性

各エージェントの行動に多様性を導入しようという試みは, いくつかの論文で実験, 検証が行われている．その 1 つとして, Jiang らによる Emergence of Individuality (EOI) [10] がある．この手法において, エージェントは実際の環

©2022 Information Processing Society of Japan

境から得られる報酬の他に, 内部報酬と呼ばれる報酬を加えることによってエージェントの行動を多様にすることを試みる手法である. EOI においては 2 つの環境において実験が行われた. 1 つ目の環境は Pac-Men と呼ばれる環境で, 中央の部屋に 4 つのエージェントがいる状態から始まり, 中央以外に存在する, 上下左右 4 つの部屋にあるドットを取得すればするほど高い報酬を得られる. 最も高い報酬を獲得する行動は, 4 つのエージェントがそれぞれ異なる部屋に入り, その部屋の中でドットを獲得する行動である.

2 つ目の環境は Windy Maze と呼ばれる環境である, 全体の環境は T 字のような通路があり, 2 つのエージェントが一番下の部分にいる状態から始まる. 上部分には右方向に風が吹いている状態を想定しており, どのような移動に対しても必ず右に移動するような条件がある. T 字の上部分の両端にはたどり着くことによって報酬を獲得できる場所がそれぞれ 1 つずつ存在しており, エージェントがそれぞれ異なる両端にたどり着くことによって, 最大の報酬である 2 を獲得できる.

この手法では Pac-Men, Windy Maze どちらの環境においても, 既存手法と比較して高い性能を示した. Windy Maze の環境では風がある中, T 字の上部分の両端にたどり着くできた. また, Pac-Men の環境においても, 既存手法では, 4 つのエージェントが上下左右 4 つの部屋にそれぞれ入ることはあまり見られなかった一方で, この手法ではそれぞれ異なる部屋に入り, ドットを取得した.

行動の多様性を実現しようとする他の研究として, Li らによる Celebrating Diversity in Shared Multi-Agent Reinforcement Learning (CDS) [17] がある. この手法においても, エージェントは実際の環境から得られる報酬の他に, 内部報酬を加えている.

また, この手法ではエージェントの行動が多様になりすぎる悪影響を抑制するために, 内部報酬の他に, 価値関数を複数作成する手法を採用している.

一般に強化学習では得られる報酬を最大化するような形で学習を行うため, この手法ではエージェントの軌跡とどのエージェントであるかを特定する ID との間の相互情報量を最大化するような学習が行われることになる. これにより, 異なるエージェントがそれぞれ異なる行動をとるような学習が行われることになる.

エージェントの行動を多様にする試みは多様な環境に対して機能する一方で, あまりに行動が多様になりすぎると環境から得る報酬が低下してしまう [17]. この研究の目的の 1 つとして多様な行動を促進する一方で, 高い報酬を獲得することができるような学習手法を実現することにある.

この研究では, 各エージェントが行動を決定する際に用いる価値関数を 2 つ用いる手法を採用している. 1 つ目として, 全体で共有される価値関数を導入し, 2 つ目として個々のエージェントが個別に持つ価値関数を導入する. 各エージェントに対応する価値関数は全体で共有される価値関数と個々のエージェントが持つ価値関数の和の形で表現される. 個々のエージェントが個別に持つ価値関数によってエージェントの行動が多様になることに対して, 全体で共有される価値関数を導入することによって全体の報酬を考慮した行動を行うようになる. また, エージェントが個別に持つ行動価値関数 $Q_i^I(a_i|\tau_i)$ の学習を行う際に L1 ノルムを正則化項として加えることにより, エージェントの行動が多様になりすぎることを抑制した.

この手法の効果を検証するために行われた実験環境は Google Research Football (GRF) [18] と StarCraft II micromanagement (SMAC) [9] と呼ばれる 2 つの環境である.

GRF は簡易的なサッカーを行う環境であり, 各エージェントがプレイヤーに対応している. エージェントの行動はそれぞれ, 前後左右に動いたり, パスやシュートを行うものである. 本実験では, その中でも比較的簡単な環境である, 3vs1, つまりエージェントが 3 人のプレイヤーとして, 1 人のキーパーに対してゴールを入れることを目的としたタスクを対象としている. SMAC は StarCraft II を題材とした環境で, エージェントが各ユニットに対応しており, 敵ユニットを全て倒すことを目標とする, Corridor と呼ばれるタスクなどがこの実験における対象となっている.

まず GRF においては 3 つのタスクにおいて実験が行われたが, どのタスクにおいても最終的なスコアにおいて, 既存手法を上回る性能が確認された. 既存手法では全てのエージェントが 1 つのボールに対して集まってしまい, 結果としてゴールすることができないという課題があることが確認できた. 一方, この手法におけるエージェントの動作の例では, ボールを持っていないエージェントがボールの方に集まるだけではなく, ゴールの方に動いてパスを受け取り, シュートを行うことができる位置に移動していることが確認された. その結果ゴールをすることができ報酬を得ることにつながった. SMAC においては, この手法によりエージェントのうち 1 体が多数の敵ユニットを引き付け, おとりとなることによってその間に味方のユニットが敵ユニットを順番に倒すといった動きができており, 高い報酬を得ることが可能であることが確認できた.

その他の関連研究として, Franzmeyer らによる研究 [19] では, 他のエージェントが将来到達できることのできる状態の数を最大化するような行動を選択させることによって, 外部報酬なしで利他的な行動を学習できることを示した.

4. 提案手法

これまでの研究においては, エージェントそれぞれが異なる行動を取ることに対して, 環境から得られる外部報酬とは別の形で内部報酬として高い報酬を与えることによりエージェント間の多様性を促進するものであった. 一方で, 本研究では, より明示的な形でエージェントを特徴づける

©2022 Information Processing Society of Japan

手法を提案する．つまり, 特定のエージェントは自身の報酬を優先する一方で, 明示的に作成した補助的なエージェントは自身の報酬よりもむしろ他のエージェントが報酬を獲得することを助けるような行動を取るような学習を行うような手法を検討する．

多くのマルチエージェント強化学習の設定ではあるエージェントが報酬を獲得した際, その報酬をエージェント全体が共有して獲得できるようになっている手法を採用しており, 本稿においてもベースラインとして, この手法を採用している. 一方, 本手法においては, 特定のエージェントが報酬を獲得する際には全エージェント間に報酬を共有するが, 補助的なエージェントとして設定したエージェントが報酬を取得する行動を行った際には, 補助的なエージェントを含め, どのエージェントも報酬を全く獲得しないという手法を提案する．

これにより, 補助的なエージェントは自身が報酬を獲得するよりも, 一般のエージェントが報酬を獲得する方が得られる報酬が高くなる．したがって, 補助的なエージェント自身が直接報酬を獲得するよりもむしろ, 一般のエージェントが報酬を獲得することを手助けするような行動を行うことが促進される．

実際のエージェントが報酬 r を獲得する行動を選択した時の全体のエージェントへ共有される報酬を r_s とすると,

$$r_s = \alpha r$$

一般のエージェントが報酬を獲得するとき $\alpha = 1$ となる．ベースラインでは全てのエージェントを一般のエージェントとして想定しているため, 常に $\alpha = 1$ となり, 報酬がそのまま全体に共有される．

提案手法においては一般のエージェントと補助的なエージェントがどちらも存在しているが, そのうち補助的なエージェントが報酬を獲得するとき, $\alpha = 0$ と設定しており, 全く報酬が共有されない．

5. 実験環境

実験に用いる環境としては PressurePlate と呼ばれる環境と simple_tag と呼ばれる 2 つの環境を用いた．PressurePlate においては提案手法と MADDPG, ランダムという 3 つのアルゴリズムを比較し, simpla_tag においては提案手法と MADDPG との間の比較を行った．

5.1 PressurePlate

PressurePlate の環境の例を図 1 に示す．PressurePlate は複数のエージェント, プレート, ドア, ゴール, 壁からなるグリッドで示される環境であり, エージェントがプレートの上に辿り着くことによって, 対応するドアが開く．また, エージェントがプレートの上から離れると, ドアが閉まる動作をする．ゴールはドアの先に設計されているため, エージェントがゴールに辿り着くためにはあるエージェントがプレートを踏むことによってドアを開け, その他のエージェントがドアの先に進み, ゴールに辿り着くといった複数のステップを踏む必要がある．PressurePlate におけるエージェントの行動は離散的であり, それぞれ上に動く, 下に動く, 右に動く, 左に動く, 何もしないといった 5 つの行動のうち 1 つの行動を選択する．各エージェントは自身の周囲の情報を観測情報として取得する．本実験においては自身から 2 マス離れた地点までの情報を観測として扱う．つまりエージェントは自身を中心とする 5×5 の範囲を観測として獲得できる．また, PressurePlate における報酬としてはゴールに辿り着くことによって +10 の報酬を獲得し, ゴールに辿り着かなかった場合は 1 タイムステップごとに -0.1 の報酬を獲得するものとして設計した．1 エピソードにおけるタイムステップは最大 100 と設定しているため, エージェントの報酬は +10 から -10 の範囲で変動する．エージェントが獲得した報酬が高いほど優れた手法であるといえる．

本実験では, エージェントが 2 つ存在し, プレート, ゴールはそれぞれ 1 つ, プレートに対応して開くドアが 2 つ存在する環境である PressurePlate_2 と, エージェントが 3 つ存在し, プレート, ゴールはそれぞれ 2 つ, 各プレートに対応して開くドアがそれぞれ 2 つずつ存在する環境である PressurePlate_3 という 2 つの環境を用いて提案手法の効果の検証を行った．各環境の様子は付録 A に示す．提案手法においては, この環境において, 補助的エージェントはプレートを踏むことによって, ドアを開ける役割を果たし, 自身はゴールに辿り着くことはできないが, 他のエージェントがゴールに辿り着くことをサポートするような行動をすることを期待するものである．

5.2 Simple_tag

Simple_tag は Multi-Agent Particle Environment (MPE) [20] に含まれている環境の 1 つである．simple_tag は, 図 2 に示すような環境であり, 緑の円形で示される 1 つのエージェントを, 赤の円形で示される複数 のエージェントが追いかける環境である．本環境においては赤の円形で示されるエージェントは 3 つ存在する．また, その他の設置物として動かない黒色で示される障害物が 2 つ存在している．緑のエージェントは赤のエージェントと接触すると負の報酬を受け取る一方で, 赤のエージェントは緑のエージェントと接触することにより, 正の報酬を受け取る．

エージェントは赤と緑それぞれによって動くことのできる速さが異なっており, 逃げる側である緑のエージェントは, 追いかける側の赤のエージェントより速く動くことができる．したがって, 赤のエージェントは単純に緑のエージェントを追いかけるだけでは捕まえることができず, より高い報酬を獲得するためには障害物を利用したり, 複数

©2022 Information Processing Society of Japan

のエージェント同士で追いかけ方を工夫したりする必要がある．また, 緑のエージェントが画面から出て移動しないように, 緑のエージェントが画面の外へ移動することに対してはペナルティとして負の報酬が与えられる．

エージェントの行動空間は赤, 緑どちらの場合であっても同じであり, それぞれのエージェントが, 動かない, 右に動く, 左に動く, 上に動く, 下に動くといった 5 つの行動が可能である．また観測できる状態としては, 自身の速度, 位置, 障害物が存在する相対的な位置, 他のエージェントが存在する相対的な位置, 他のエージェントの速度が与えられる．また 1 エピソードにおけるタイムステップは最大 25 と設定した．

本環境において学習するエージェントは大きく分けて 2 種類存在する．図 2 に示される緑色の円形で示されるエージェント (以下ターゲットエージェントと呼ぶ) と, 赤色の円形で示されるエージェント (以下チェイサーエージェントと呼ぶ) の 2 種類である．このうち, 提案手法の検証に用いるエージェントはチェイサーエージェントである．チェイサーエージェントは全部で 3 つ存在するが, そのうち 1 つのエージェントを一般のエージェントとし, それ以外の 2 つのエージェントを補助的なエージェントとして設定する．ここではターゲットエージェントの学習アルゴリズムはチェイサーエージェントの学習アルゴリズムに関わらず全て MADDPG を用いて学習を行った．

手法を評価する際に注意する点として, 提案手法においては方策を学習する際にはエージェントが環境から報酬を受け取った際, 報酬の共有のされ方についてそれぞれ変化させていたが, 本実験結果を示す際には手法によらず, 報酬を統一することで評価指標として用いた．具体的には, チェイサーエージェントのうちどのエージェントがターゲットエージェントと接触したとしても, 報酬を獲得するとして扱い, またターゲットエージェントが画面の外に出てしまう場合もチェイサーエージェントは追いかけることが不可能となるため報酬を獲得するものとして扱った．

6. 実験結果

図 3, 4, 5 に実験結果を示す．学習は各実験環境に対してそれぞれ 20 回ずつ行い、実線として報酬の中央値, 色付きの領域として標準偏差を示している．PressurePlate の結果を示した図 3, 4 を見ると, 提案手法によりベースライン手法である MADDPG やランダムと比較して高い性能を示していることが確認できる．ランダム手法では PressurePlate_2, PressurePlate_3 どちらの環境でも −10 の値を取っていることから, ランダム手法ではゴールに辿り着けていないことが分かる．またベースライン手法である MADDPG においても PressurePlate_3 の環境ではほぼ −10 の値を取っており, ほとんどゴールできていないことが確認できる．このような環境においても提案手法によりゴールに辿り着くことができる方策を学習できることが示された．

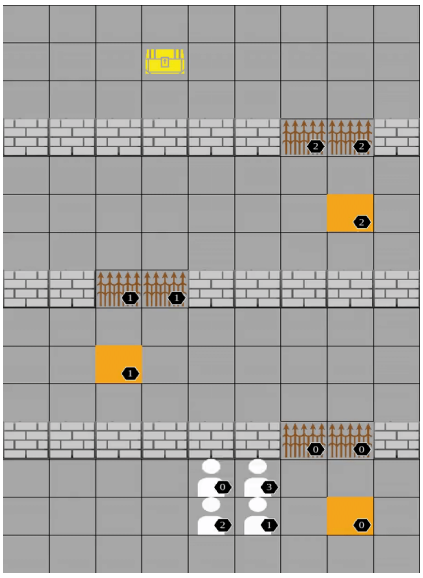

図 1　PressurePlate の環境の例

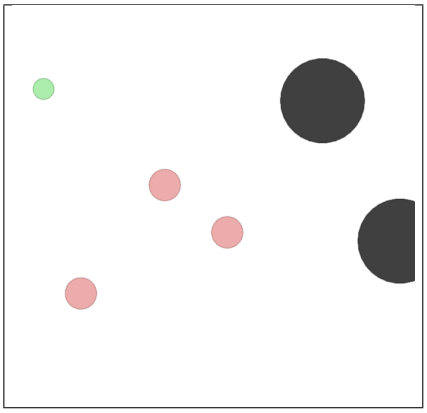

図 2　simple_tag の環境の例

一方 simple_tag の環境における結果が示された図 5 を見ると, 提案手法では MADDPG と比較してほとんど報酬を獲得できていないことが確認される．つまり simple_tag において提案手法ではターゲットエージェントを捕まえるといった行動を行う方策を獲得できなかったことが示されている．

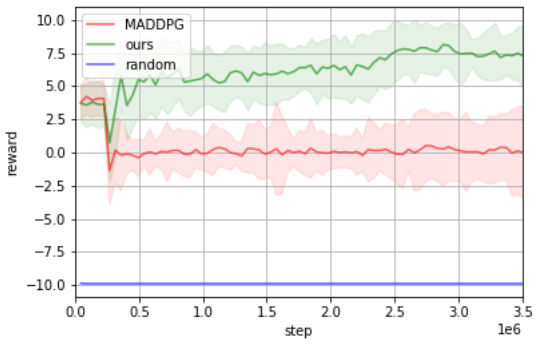

図 3　PressurePlate_2 の環境における結果

©2022 Information Processing Society of Japan

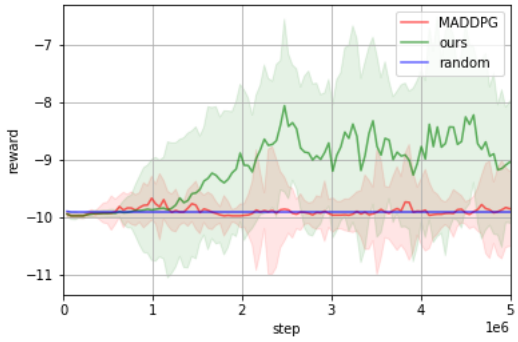

図 4　PressurePlate_3 の環境における結果

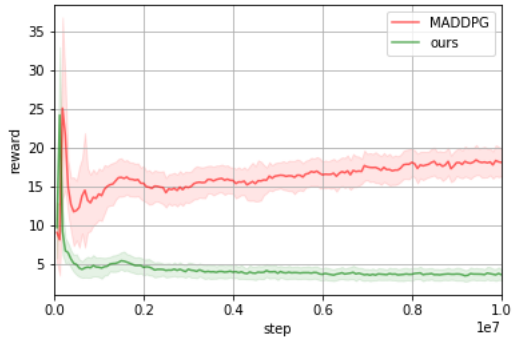

図 5　simple_tag の環境における結果

7.　考察

本稿ではマルチエージェント強化学習環境においてエージェント間の報酬の共有のされ方に着目し, 補助的なエージェントと一般のエージェントを区別することを提案した. その結果, PressurePlate と simple_tag においてベースライン手法と異なる動作をすることが確認された. PressurePlate においては補助的なエージェントがプレートを踏むという役割を明確に果たすことができるようになったことにより, 全体として他のエージェントがゴールに到達しやすくなる効果があったと推測される.

一方で simple_tag の環境では, 本来補助的なエージェントがターゲットエージェントを囲い込むなどといった行動により, 他のエージェントがターゲットエージェントを捕まえやすくなるといった状況を想定していた. しかし, 結果を見ると提案手法では上手くターゲットエージェントを捕まえることができていなかった. これは simple_tag の環境においては, ターゲットエージェントを囲い込むといった補助的な行動は, ターゲットエージェントに近づくといった意味においてエージェントを捕まえるといった行動と類似点が存在する. したがって, 補助的なエージェントが報酬を獲得しないような報酬設計を行った本手法においてはターゲットエージェントに近づくといった行動が阻害されてしまい, 結果として高い報酬を獲得することに繋がらなかったと推測される.

結果として 2 つの環境における提案手法の性能の違いは, 補助的な行動と考えられる行動が, 最終的に報酬を獲得するといった行動とどれほど異なる行動であるかという環境そのものの違いにあると考えられる. PressurePlate ではプレートを踏むといった行動がゴールに辿り着くという行動と全く異なることから上手く役割を分担するように学習が可能となった反面, simple_tag ではターゲットエージェントを囲い込むといった行動がターゲットエージェントを捕まえるという行動と, ターゲットエージェントに近づく必要があるという観点で類似点があることから上手く学習ができなかったと考えられる.

8.　まとめ

本研究ではマルチエージェント強化学習に用いられるアルゴリズムはしばしば複数のエージェントを特に区別せずに学習を行うことが多く, 複数のエージェントが同じような行動をとってしまう場合があるといった問題点に着目し, 各エージェントを区別して扱うために, 自身の報酬よりむしろ他者の報酬を獲得することを目指す補助的なエージェントを導入することを提案した. 実験環境としては PressurePlate と simple_tag という大きく分けて 2 つの環境において実験を行った. その結果 PressurePlate の環境においては提案手法により既存手法と比較して高い性能を示すことができることを確認した.

一方で simple_tag においては既存手法を大きく下回る性能を示したことから本手法が任意の環境に対して適応することが難しいという点も確認された. 以上を踏まえ, 今後の課題としてはより多様な環境において高い性能を示すアルゴリズムを提案することが求められる.

参考文献

[1] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin Riedmiller, Andreas K. Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, February 2015.

[2] David Silver, Aja Huang, Christopher Maddison, Arthur Guez, Laurent Sifre, George Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, Sander Dieleman, Dominik Grewe, John Nham, Nal Kalchbrenner, Ilya Sutskever, Timothy Lillicrap, Madeleine Leach, Koray Kavukcuoglu, Thore Graepel, and Demis Hassabis. Mastering the game of go with deep neural networks and tree search. *Nature*, 529:484–489, 01 2016.

[3] Adrià Puigdomènech Badia, B. Piot, Steven Kapturowski, P. Sprechmann, Alex Vitvitskyi, Daniel Guo, and Charles Blundell. Agent57: Outperforming the atari hu-

©2022 Information Processing Society of Japan

man benchmark. In *International Conference on Machine Learning*, 2020.

[4] Frederik Ebert, Chelsea Finn, Sudeep Dasari, Annie Xie, Alex Lee, and Sergey Levine. Visual foresight: Model-based deep reinforcement learning for vision-based robotic control. *arXiv preprint arXiv:1812.00568*, 2018.

[5] K. Zhang, Zhuoran Yang, and Tamer Başar. Multi-agent reinforcement learning: A selective overview of theories and algorithms. *ArXiv*, abs/1911.10635, 2019.

[6] Chao Yu, Akash Velu, Eugene Vinitsky, Yu Wang, Alexandre M. Bayen, and Yi Wu. The surprising effectiveness of mappo in cooperative, multi-agent games. *ArXiv*, abs/2103.01955, 2021.

[7] Zhen Zhang, Dongqing Wang, and Junwei Gao. Learning automata-based multiagent reinforcement learning for optimization of cooperative tasks. *IEEE Transactions on Neural Networks and Learning Systems*, 32:4639–4652, 2021.

[8] Xingyu Wang and Diego Klabjan. Competitive multi-agent inverse reinforcement learning with sub-optimal demonstrations. In *International Conference on Machine Learning*, 2018.

[9] Mikayel Samvelyan, Tabish Rashid, C. S. D. Witt, Gregory Farquhar, Nantas Nardelli, Tim G. J. Rudner, Chia-Man Hung, Philip H. S. Torr, Jakob N. Foerster, and Shimon Whiteson. The starcraft multi-agent challenge. *ArXiv*, abs/1902.04043, 2019.

[10] Jiechuan Jiang and Zongqing Lu. The emergence of individuality. In *International Conference on Machine Learning*, pages 4992–5001. PMLR, 2021.

[11] Kyunghwan Son, Daewoo Kim, Wan Ju Kang, David Earl Hostallero, and Yung Yi. Qtran: Learning to factorize with transformation for cooperative multi-agent reinforcement learning. *ArXiv*, abs/1905.05408, 2019.

[12] Yue Wang, Yao Wan, Chenwei Zhang, Lixin Cui, Lu Bai, and Philip S. Yu. Competitive multi-agent deep reinforcement learning with counterfactual thinking. *2019 IEEE International Conference on Data Mining*, pages 1366–1371, 2019.

[13] Michael L. Littman. Markov games as a framework for multi-agent reinforcement learning. In *Proceedings of the Eleventh International Conference on International Conference on Machine Learning*, page 157–163, 1994.

[14] Ryan Lowe, Yi Wu, Aviv Tamar, Jean Harb, P. Abbeel, and Igor Mordatch. Multi-agent actor-critic for mixed cooperative-competitive environments. In *Neural Information Processing Systems*, 2017.

[15] David Silver, Guy Lever, Nicolas Manfred Otto Heess, Thomas Degris, Daan Wierstra, and Martin A. Riedmiller. Deterministic policy gradient algorithms. In *International Conference on Machine Learning*, 2014.

[16] Vijay R Konda and John N Tsitsiklis. Actor-critic algorithms. In *Advances in neural information processing systems*, pages 1008–1014, 2000.

[17] Chenghao Li, Tonghan Wang, Chengjie Wu, Qianchuan Zhao, Jun Yang, and Chongjie Zhang. Celebrating diversity in shared multi-agent reinforcement learning. *Advances in Neural Information Processing Systems*, 34, 2021.

[18] Karol Kurach, Anton Raichuk, Piotr Stanczyk, Michal Zajac, Olivier Bachem, Lasse Espeholt, Carlos Riquelme, Damien Vincent, Marcin Michalski, Olivier Bousquet, and Sylvain Gelly. Google research football: A novel reinforcement learning environment. *ArXiv*, abs/1907.11180, 2020.

[19] Tim Franzmeyer, Mateusz Malinowski, and Joao F. Henriques. Learning altruistic behaviours in reinforcement learning without external rewards. In *International Conference on Learning Representations*, 2022.

[20] Ryan Lowe, Yi Wu, Aviv Tamar, Jean Harb, Pieter Abbeel, and Igor Mordatch. Multi-agent actor-critic for mixed cooperative-competitive environments. *Neural Information Processing Systems*, 2017.

付　　録

A　実験環境

本稿で実験を行った環境である PressurePlate において，エージェントの数やオブジェクトの配置などを変更した 2 つの環境について実験を行った．図 A·1 に PressurePlate_2 の環境を，図 A·2 に PressurePlate_3 の環境をそれぞれ示す．

	G			
W	W	D	D	W
A1	P			
A2				

図 A·1　PressurePlate_2 の環境．

	G			
W	W	D	D	W
	P			
W	W	D	D	W
A3	P			
A2	A1			

図 A·2　PressurePlate_3 の環境．

A1, A2, A3 はそれぞれエージェント，W は壁，G はゴール，D はドア，P はプレートの位置を示している．また PressurePlate_3 の環境ではプレートが二か所存在しているが，上から 4 列目のプレートが 3 列目のドアに，上から 7 列目のプレートが 6 列目のドアにそれぞれ対応している．またどちらの環境においてもエージェントの観測は変化せず，自身を中心とした 5×5 の範囲を観測として獲得できる．

©2022 Information Processing Society of Japan