Vol.2022-ICS-205 No.12
2022/2/21

マルチエージェント強化学習の差分報酬近似手法における推定報酬の学習の改良

中嶋 亮太[1,a)]　松井 俊浩[2]

概要：差分報酬は，各エージェントの貢献度に応じた報酬であり，協調問題におけるマルチエージェント強化学習の改善のために用いられる．しかし，環境全体の情報や大域的報酬の計算式が未知の状況では，差分報酬を計算することができない．従来研究では，大域的報酬関数がブラックボックスの状況で，各エージェントの局所的な情報と関連付けて大域的報酬を推定する関数を学習し，差分報酬を近似する手法が提案されている．従来手法では推定報酬の学習にニューラルネットワークを用いているが，ニューラルネットワークの更新部分に改良の余地があると考えられる．本研究では，従来手法における推定報酬の学習に対して，一般的な推定手法で使用されている改良手法，学習データの標準化とミニバッチ勾配降下法を適用する．ベンチマーク問題を用いた実験により提案手法を評価し，推定報酬の学習の改良手法により，比較的小規模な環境において，マルチエージェント強化学習系全体の獲得報酬が改善されることを示す．

1. はじめに

マルチエージェントシステムは，自律的な主体であるエージェントが複数存在する系であり，交通管制や資源配分，複数ロボットの制御など現実世界の多くの問題をモデル化することができる [1], [2]．協調的なマルチエージェントシステムでは，各エージェントが共通の目的を達成するために最適な行動を選択する必要があり，最適な行動を学習するために強化学習が用いられる [3]．強化学習において，環境内のエージェントは，観測した状態に応じた行動を決定し，行動に応じた報酬を環境から受け取ることで最適な行動を学習する．環境内に複数のエージェントが存在するマルチエージェント強化学習では，環境全体の状態と全エージェントの行動に応じたグローバル報酬が各エージェントに与えられるが，グローバル報酬は個々のエージェントの状態と行動に応じた報酬ではない．そのため，グローバル報酬では各エージェントの環境に対する貢献度が不明であり，環境全体の利益に貢献していないエージェントに対しても同じ報酬が与えられるという問題がある．

この問題を解決するために，各エージェントに差分報酬（Difference Rewards）を与える方法が提案されている．差分報酬は，グローバル報酬の値と，報酬の評価対象であるエージェントの影響を取り除いた場合のグローバル報酬の値との差分をとることにより計算される．差分報酬を受け取ることで，各エージェントは自身の行動が環境に与える貢献度を知ることができ，よりよい行動を学習しやすくなる．そのため，差分報酬を用いたマルチエージェント強化学習は，さまざまな問題で有用性が示されている [4]．しかし，環境全体の情報やグローバル報酬の計算式が未知であるマルチエージェント強化学習においては，評価対象エージェントの影響を取り除いた報酬を計算することができず，差分報酬を計算することが困難である．

この差分報酬の課題に対し，従来研究 [5] では，個々のエージェントの状態と行動からグローバル報酬の値を推定する関数を学習することで差分報酬を近似する手法が提案されている．従来手法では推定報酬を学習するためにニューラルネットワークを用いているが，ニューラルネットワークの更新部分に改良の余地があると考えられる．本研究では，一般的なニューラルネットワークで用いられる学習データの標準化とミニバッチ勾配降下法を適用することにより，従来手法で用いられている推定報酬の学習を改良する．ベンチマーク問題を用いた実験により，推定報酬の学習に改良手法を適用した提案手法により，比較的小規模な問題において，マルチエージェント強化学習系全体の獲得報酬が改善されることを示す．

2. 研究背景

ここでは，マルチエージェント強化学習と差分報酬，およびその課題について説明する．

[1] 名古屋工業大学工学部情報工学科
[2] 名古屋工業大学大学院工学研究科情報工学専攻
[a)] r.nakajima.979@stn.nitech.ac.jp

2.1 マルチエージェント強化学習

シングルエージェントによる強化学習では，エージェントが状態観測から行動を決定し，行動に対する環境からの報酬受け取る．エージェントは獲得できる報酬の最大化を目的とすることで，行動の系列である方策を学習する．一方，協調的なマルチエージェント強化学習では，環境内に複数のエージェントが存在し，共通の目的を達成するためにより良い方策を学習する．環境内の各エージェントには環境全体の状態と全エージェントの行動に応じたグローバル報酬が与えられるが，グローバル報酬は個々のエージェントの状態と行動に応じた報酬ではないため，各エージェントの系全体に対する貢献度が不明である．各エージェントの貢献度が不明なグローバル報酬を全エージェントに与えるということは，系全体の利益に貢献していないエージェントに対しても同じ値の報酬が与えられることになるため，そのようなエージェントの方策が改善されにくく，マルチエージェント強化学習の課題の 1 つである [6].

2.2 差分報酬（Difference Rewards）

このような課題の解決策の 1 つとして，各エージェントに差分報酬（Difference Rewards）を与える方法が提案されている．差分報酬は次のように定義され，各エージェントの貢献度に応じた報酬を計算することができる．

$$D_i(s,a) = G(s,a) - G(s_{-i} \cup s_{c,i}, a_{-i} \cup a_{c,i}) \qquad (1)$$

- $D_i(s,a)$ はエージェント i が受け取る差分報酬
- $G(s,a)$ は環境全体のグローバル報酬
- s, a は環境全体の状態と全エージェントの行動
- s_{-i} はエージェント i の影響を除いた環境全体の状態
- a_{-i} はエージェント i 以外すべてのエージェントの行動
- $s_{c,i}, a_{c,i}$ はエージェント i の状態と行動を置換する値

式 (1) において，第 2 項ではエージェント i の影響を除いた場合のグローバル報酬を計算している．この値とグローバル報酬との差分をとることにより，系全体の利益に対するエージェント i によるグローバル報酬の増減が計算される．差分報酬を計算しエージェントに与えることで，各エージェントは自らの貢献度を知ることができるため，差分報酬を用いたマルチエージェント強化学習は，航空交通管制や複数ロボットの制御，混雑ゲームなどさまざまなマルチエージェントの協力問題で有用性が示されている [7], [8].

2.3 差分報酬の課題

差分報酬はさまざまな問題で優れた性能が示されている手法だが，問題設定によっては計算できない場合がある．具体的には，各エージェントが環境全体の状態情報を完全には入手することができず，グローバル報酬の計算式が未知であるという問題設定において，式 (1) の第 2 項 $G(s_{-i} \cup s_{c,i}, a_{-i} \cup a_{c,i})$ を計算することができない．一般的なマルチエージェント強化学習の問題設定では，各エージェントがグローバル報酬の値を与えられるが，環境全体の情報とグローバル報酬の計算式を利用することができないため，差分報酬を利用するためには環境全体を把握し式 (1) の第 2 項を計算することができる中央アルゴリズムを導入する必要がある．しかし，そのような中央アルゴリズムを導入できない問題設定では，差分報酬を利用することができない．

3. 従来研究

問題設定により差分報酬が計算できない場合があるという課題に対し，従来研究 [5] では個々のエージェントが利用できる局所的な情報と環境から与えられるグローバル報酬から差分報酬を近似する手法が提案されている．ここでは，差分報酬を近似する手法と，従来研究で使用されていたベンチマーク問題と方策の学習方法について説明する．

3.1 差分報酬の近似

問題設定によっては差分報酬を計算することができないという課題に対し，従来研究では差分報酬を近似する手法が提案されている．近似された差分報酬は次のように定義される．

$$\hat{D}_i(s,a) = G(s,a) - \hat{G}_i(s_{c,i}, a_{c,i}) \qquad (2)$$

ここで，$\hat{D}_i(s,a)$ はエージェント i の差分報酬を近似した値である．2.3 節で述べたように，差分報酬が計算できない原因は，式 (1) の第 2 項である $G(s_{-i} \cup s_{c,i}, a_{-i} \cup a_{c,i})$ を計算できないことであり，この値を近似することで差分報酬を近似することができる．式 (2) の第 2 項における $\hat{G}_i$ は，エージェント i の状態 s_i と行動 a_i を入力として，グローバル報酬の値を推定する関数である．式 (2) では，$\hat{G}_i$ の入力を $s_{c,i}, a_{c,i}$ にすることで $G(s_{-i} \cup s_{c,i}, a_{-i} \cup a_{c,i})$ を近似している．推定報酬 $\hat{G}_i$ は各エージェントが保持しており，エージェントの局所的な状態観測と行動からグローバル報酬を推定することにより，環境全体の情報とグローバル報酬の計算式が利用できないという制約下で，個々のエージェントがグローバル報酬の値を計算できるようにする．$\hat{G}_i$ の学習については 3.4 節で示す．

3.2 ローバー問題

従来研究では，連続的な状態行動空間を扱うベンチマーク問題としてローバー問題 [9], [10] を用いている．この問題により，未知の環境で限られた状態観測をもとに，それぞれのエージェントが向かう方向をどの程度分担できているかを調べることができる．本研究においても，手法を評価するためのベンチマーク問題としてローバー問題を用いるため，ここではローバー問題の環境やグローバル報酬，

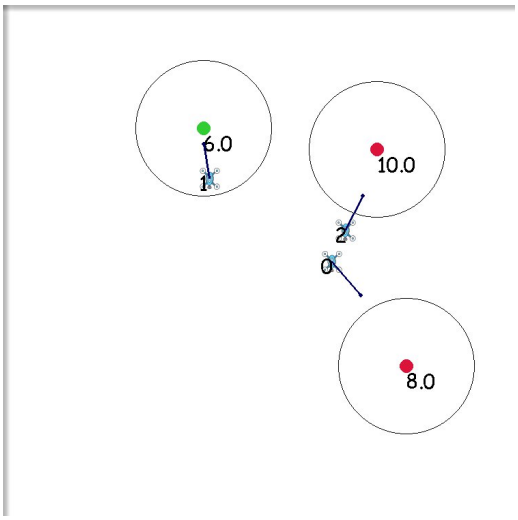

図 1 ローバー問題（青:エージェント，緑または赤:POI，POI 横の数値:POI の価値）

エージェントの状態観測や行動について説明する．

ローバー問題は，**図 1** のように複数の地点（POI）が点在する 2 次元平面上で，複数のエージェントが動き回り POI を観測する問題である．各 POI にはそれぞれ価値 V が設定されており，エージェントが POI に近づくと，エージェントと POI の距離に応じた観測値が得られる．このドメインにおける距離情報は式 (3) のようにユークリッド距離の 2 乗を使用し，この値が 0 をとらないように最小値 δ_{min} を設定する．

$$\delta(x,y) = max\{||x-y||, \delta^2_{min}\} \tag{3}$$

ローバー問題のグローバル報酬は次の式により計算される．エージェントの共通の目的は，1 エピソードで得られるグローバル報酬の値を最大化することである．

$$G = \sum_j \frac{V_j}{min_i \delta(L_j, L_i)} \tag{4}$$

ここで，V_j は POIj の価値，L_j は POIj の位置，L_i はエージェント i の位置である．式 (4) からわかるように，グローバル報酬は各 POI に対して最も近いエージェントとの距離を考慮する．また，グローバル報酬は各ステップで計算され、エージェントが POI を観測し続けるとエージェントに継続して報酬が与えられる．

エージェントは象限ごとに POI と他のエージェントの情報を取得し状態を観測する．エージェント i の，象限 q における POI に関する観測値 $s_{1,q,i}$ と，他のエージェントに関する観測値 $s_{2,q,i}$ はそれぞれ以下のように定義される．

$$s_{1,q,i} = \sum_{j \in I_q} \frac{V_j}{\delta(L_j, L_i)} \tag{5}$$

$$s_{2,q,i} = \sum_{i' \in N_q} \frac{1}{\delta(L_{i'}, L_i)} \tag{6}$$

ここで，I_q は象限 q 内にある POI の集合であり，N_q は象限 q 内にいるエージェントの集合である．エージェントはこの状態観測値から x 方向と y 方向それぞれの変異を行動として決定する．

3.3 共進化アルゴリズム

ローバー問題でエージェントが方策を学習するために，従来研究では共進化アルゴリズム [9] を使用している．共進化アルゴリズムでは，各エージェントが，各ステップで状態観測値から行動を決定するための方策ニューラルネットワーク（NN）を k 個保持している．エージェントは各エピソード開始時に，方策 NN を $2k$ 個に増加させる．各エージェントが $2k$ 個からランダムに 1 つ選択し，シミュレーションを実行する．方策を評価するための報酬としてグローバル報酬，差分報酬，または近似した差分報酬を用い，各方策 NN は 1 エピソードで得られた報酬の合計を評価値とする．エピソード終了時に，$2k$ 個の方策 NN から評価値が最も高いもの，もしくはランダムに選択されたものを k 個選択する．エピソード終了後のテスト段階では，各エージェントが保持する方策 NN からそれぞれ最も評価値が高いものを選択しシミュレーションを実行する．

3.4 従来手法における推定報酬の学習

ローバー問題におけるエージェントの状態観測値と行動は連続値であるため，従来研究では報酬の推定に中間層が 1 層のニューラルネットワーク（NN）を用いている．これにより，共進化アルゴリズムにより方策を学習する NN とは別に，報酬を推定するための NN を各エージェントは保持することになる．ここでは，報酬を推定する NN の学習について説明する．

推定報酬により近似した差分報酬を用いた共進化アルゴリズムを**図 2** に示す．図 2 の 4 行目で報酬を推定するためのニューラルネットワークを初期化している．14 行目から 21 行目で各エピソードにおける各ステップでの処理を示している．15 行目と 16 行目で各エージェントが状態を観測し行動を決定する．17 行目で状態遷移後のグローバル報酬を受け取り，18 行目で推定報酬を更新する．18 行目の更新処理では，状態遷移前に観測された状態 s_i とそれに応じて決定された行動 a_i を入力として $\hat{G}_{i,j}(s_i, a_i)$ を計算し，状態遷移後に環境から与えられたグローバル報酬 $G(s,a)$ との誤差 $G(s,a) - \hat{G}_{i,j}(s_i, a_i)$ を求め，誤差逆伝播法により報酬推定 NN を更新する．

従来手法による推定報酬の更新は各ステップで得られたデータをその場で利用しているが，一般的なニューラルネットワークの学習では，まとまったデータを学習しやすいように事前に処理し，パラメータの更新時には複数のデータを同時に用いる場合が多い．このようなことを考慮すると，各ステップでパラメータを更新するのではなく，

Algorithm 1 $\hat{D}_i(s,a)$ を適用した共進化アルゴリズム

```
1: k 個の方策 NN を持つ N 体のエージェントを初期化
2: for Agent i do
3:     for NeualNetwork j do
4:         報酬推定 NN Ĝ_{i,j}(s_i, a_i) を初期化
5:     end for
6: end for
7: for Episode do
8:     for Agent do
9:         k 個の方策 NN を複製
10:        複製した方策 NN を変異させる
11:    end for
12:    for j = 1 → 2k do
13:        各エージェントがランダムに方策 NN を選択
14:        for Timestep do
15:            各エージェントが状態 s_i を観測
16:            各エージェントが行動 a_i を決定し状態遷移
17:            全エージェントが G(s,a) を獲得
18:            G(s,a) − Ĝ_{i,j}(s_i, a_i) を誤差として Ĝ_{i,j}(s_i, a_i) を更新
19:            D̂_{i,j}(s,a) = G(s,a) − Ĝ_{i,j}(c_{s,i}, c_{a,i})
20:            D̂_{i,j}(s,a) により方策 NN の評価値を更新
21:        end for
22:    end for
23:    for Agent do
24:        ϵ-Greedy 法により k 個の方策 NN を選択
25:    end for
26: end for
```

図 2 従来手法による推定報酬の更新

1 エピソードの終了時にまとめて更新することで，ニューラルネットワークの学習で用いられるさまざまな手法を適用することができ，推定報酬の学習が改善されると考えられる．

4. 提案手法

報酬を推定するニューラルネットワークを 1 エピソードの終了時に更新することで，学習に必要なデータをまとめることができ，一般的なニューラルネットワークの学習で用いられる手法を適用することができる．本研究では，学習データの標準化とミニバッチ勾配降下法を，従来手法における推定報酬の学習に適用する．ここでは，これら 2 つの手法の概要と従来手法への適用方法について説明する．

4.1 データの標準化

一般的なニューラルネットワークの学習では，学習データをニューラルネットワークの学習に適した値に変化させるために，学習データの前処理をする [11]．データの前処理では，データをスケーリングすることが多く，標準化はスケーリング手法の 1 つである．あるデータセットを $X = \{x_0, x_1, \ldots, x_i, \ldots\}$ とし，X の平均を μ，標準偏差を σ とすると，データセット X 内の 1 つのデータ x_i を標準化したデータ $x_{z,i}$ は次のように計算される．

$$x_{z,i} = \frac{x_i - \mu}{\sigma} \tag{7}$$

学習データの各特徴量に対して標準化をすることにより，各特徴量が平均 0 標準偏差 1 になるようにデータが変換される．特徴量間でスケールが等しくなるため，それぞれの特徴量に対するパラメータの更新幅に偏りがなく，学習が速くなるとされている．

次に，従来手法における推定報酬の学習に適用する方法を示す．報酬推定 NN の更新を各エピソード終了時にまとめ，学習データの標準化を適用したアルゴリズムは**図 3** のように表すことができる．図 2 と比較すると，図 3 では 18 行目の処理が推定報酬の更新から学習データの保存に変わり，22 行目から 25 行目の処理が追加されている．各ステップの処理が記載されている 14 行目から 21 行目では報酬推定 NN の更新をせず，18 行目で更新に必要な情報である各ステップでの s_i，a_i，$G(s,a)$ の値を学習データとして保存する．各エピソードにおける全ステップの処理を終えた 22 行目から 25 行目で，各エージェントが報酬推

Algorithm 2 従来手法における $\hat{G}_i$ の学習に対し学習データの標準化を適用した共進化アルゴリズム

```
1: k 個の方策 NN を持つ N 体のエージェントを初期化
2: for Agent i do
3:     for NeualNetwork j do
4:         報酬推定 NN Ĝ_{i,j}(s_i, a_i) を初期化
5:     end for
6: end for
7: for Episode do
8:     for Agent do
9:         k 個の方策 NN を複製
10:        複製した方策 NN を変異させる
11:    end for
12:    for j = 1 → 2k do
13:        各エージェントがランダムに方策 NN を選択
14:        for Timestep do
15:            各エージェントが状態 s_i を観測
16:            各エージェントが行動 a_i を決定し状態遷移
17:            全エージェントが G(s,a) を獲得
18:            s_i, a_i, G(s,a) を学習データとして保存
19:            D̂_{i,j}(s,a) = G(s,a) − Ĝ_{i,j}(c_{s,i}, c_{a,i})
20:            D̂_{i,j}(s,a) により方策 NN の評価値を更新
21:        end for
22:        for Agent do
23:            1 エピソードで得られた学習データを標準化
24:            G(s,a) − Ĝ_{i,j}(s_i, a_i) を誤差として Ĝ_{i,j}(s_i, a_i) を更新
25:        end for
26:    end for
27:    for Agent do
28:        ϵ-Greedy 法により k 個の方策 NN を選択
29:    end for
30: end for
```

図 3 学習データの標準化を適用した推定報酬の更新

定 NN を更新する．23 行目では，1 エピソードで得られた学習データの各特徴量を式 (7) により標準化する．24 行目では，標準化されたデータから各ステップにおける s_i，a_i，$G(s,a)$ の標準化後の値を取り出し，従来手法と同様に $G(s,a) - \hat{G}_{i,j}(s_i,a_i)$ を誤差として誤差逆伝播法により報酬推定 NN を更新する．

また，学習データの標準化に用いる平均と標準偏差は各エピソードで得られる学習データごとの値を使用するが，図 3 の 19 行目で $\hat{G}_i$ の値を出力するときの，標準化の逆変換には各エピソードで以下の式により逐次更新される平均と標準偏差を用いる．この平均と標準偏差は，それまでのエピソードで使用したすべての学習データの平均と標準偏差である．

$$\mu_{n+1} = \frac{1}{n+1}(n\mu_n + x_{n+1}) \tag{8}$$

$$\sigma_{n+1} = \sqrt{\frac{n(\sigma_n^2 + \mu_n^2) + x_{n+1}^2}{n+1} - \mu_{n+1}^2} \tag{9}$$

ここで，n はそれまでのエピソード数，μ_n と σ_n はエピソード n までの学習データの平均と標準偏差，x_{n+1} はエピソード $n+1$ で得られた学習データの平均である．

4.2 ミニバッチ勾配降下法の適用

ニューラルネットワークのパラメータは，誤差から勾配を計算し誤差が小さくなる方向に更新される．このパラメータの更新方法を勾配降下法と呼ぶ．勾配降下法には，最急勾配降下法，確率的勾配降下法，ミニバッチ勾配降下法の 3 種類ある [12]．最急勾配降下法は本稿にはあまり関係ないが，比較のために 3 種類すべての勾配降下法におけるパラメータの更新式を以下に示す．

$$\boldsymbol{w}^{(t+1)} = \boldsymbol{w}^{(t)} - \eta^{(t)} \frac{1}{m} \sum_{i \in D} \nabla E_i(\boldsymbol{w}^{(t)}) \tag{10}$$

$$\boldsymbol{w}^{(t+1)} = \boldsymbol{w}^{(t)} - \eta^{(t)} \nabla E_i(\boldsymbol{w}^{(t)}) \tag{11}$$

$$\boldsymbol{w}^{(t+1)} = \boldsymbol{w}^{(t)} - \eta^{(t)} \frac{1}{m'} \sum_{i \in B_j} \nabla E_i(\boldsymbol{w}^{(t)}) \tag{12}$$

ここで，$\boldsymbol{w}^{(t)}$ は t 回目の更新におけるニューラルネットワークのパラメータ，$\eta^{(t)}$ は t 回目の更新における学習率，$\nabla E_i(\boldsymbol{w}^{(t)})$ はデータ i におけるパラメータ $\boldsymbol{w}^{(t)}$ に対する損失関数の勾配である．また，D は全学習データの集合で m はその大きさであり，B_j はミニバッチ j の学習データ集合であり m' はそのバッチサイズである．

式 (10) は最急勾配降下法の更新式である．最急勾配降下法はバッチ勾配降下法とも呼ばれ，すべての学習データについて誤差を計算しその平均の勾配でパラメータを更新する手法である．学習データすべてに対しての勾配を計算するため解への収束が速く学習が安定する一方で，局所解を回避できないという特徴がある．式 (11) は確率的勾配降下法の更新式であり，ランダムに選択されたデータ 1 つに対して勾配を計算しパラメータを更新する手法である．パラメータの更新に 1 つのデータに対する勾配しか考慮しないため解への収束が遅く学習が不安定であるが，局所解を回避できる可能性がある．式 (12) に示したミニバッチ勾配降下法は，学習データをミニバッチと呼ばれるデータ群に分割し，各ミニバッチ内のデータに対して誤差を計算しその平均の勾配でパラメータを更新する．ミニバッチ勾配降下法は，最急勾配降下法とミニバッチ勾配降下法の中間的な手法であるため，2 つの手法のメリットを持ち合わせており，多くの深層学習において使用される勾配降下法である [13]．

従来手法における報酬推定 NN の更新は，各ステップで得られた 1 つのデータを使って更新しているので，確率的勾配降下法に近い学習だといえる．確率的勾配降下法は解への収束が遅く学習が不安定であるという特徴があるため，ミニバッチ勾配降下法を適用することにより報酬推定 NN の収束が速くなり学習が安定することで，マルチエージェント強化学習系全体の結果が向上すると考えられる．ミニバッチ勾配降下法の適用は，4.1 節で示した図 3 のアルゴリズムにおける 24 行目の処理を変更することでできる．図 3 の 24 行目において，4.1 節では標準化されたデータを 1 つずつ取り出し式 (11) によりパラメータを更新していたが，標準化された 1 エピソード分の学習データをミニバッチとして考え式 (12) によりパラメータを更新するように変更する．これにより，1 エピソードで得られた学習データに対する平均の勾配で報酬推定 NN を更新することができ，ミニバッチ勾配降下法に近い学習が可能になる．

5. 実験と考察

本研究では従来手法と提案手法を，3.2 節で示したローバー問題と 3.3 節で示した共進化アルゴリズムを用いて比較した．ここでは，実験で用いたローバー問題と共進化アルゴリズムの設定と，2 つの提案手法を導入した場合の実験結果を提示する．

5.1 実験設定

実験で使用するローバー問題では，15 ユニット ×15 ユニットの 2 次元平面環境に 3 体のエージェントと 3 つの POI が存在する環境設定とした．各エージェントが持つ方策 NN とそれに対応する報酬推定 NN の個数をそれぞれ 25 とした．1 回の実験におけるエピソード数は 3000 とし，各エピソードの全ステップ数は 25 とした．エージェントは 1 ステップで上下左右の方向に最大で 1 ユニット分移動することができる．報酬推定 NN における中間層のユニット数は 16 に設定した．エージェントと POI の位置，POI の価値をそれぞれランダムに設定することにより生成された 10 個の問題領域を用いて実験し，10 問における平均獲得報酬の推移を比較した．ローバー問題の実装は [14] のプロ

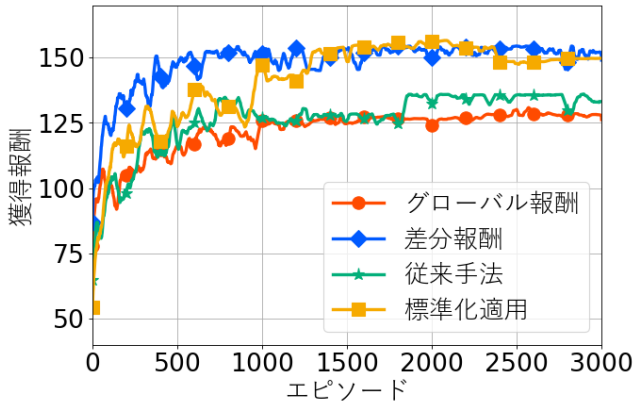

図 4 学習データの標準化を適用した実験の平均獲得報酬の推移

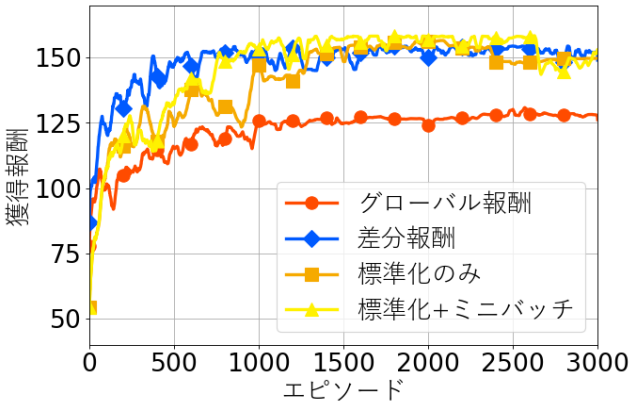

図 5 ミニバッチ勾配降下法を適用した実験の平均獲得報酬の推移

表 1 問題ごとの獲得報酬の比較

問題番号	グローバル報酬	差分報酬	従来手法	標準化適用
0	180	240	162.64	184.56
1	90	130	90	121.13
2	96.38	96.37	80.96	170
3	138.13	134.55	180	143.65
4	130	130	130	130
5	80	72.09	102.11	82.31
6	140	200	140	200
7	117.27	130	130	130
8	150	230	155.42	174.87
9	160	162.67	160	160
平均	128.17	152.57	133.11	149.65

グラムを利用し，その他の設定はローバー問題を使用した従来研究 [5], [9] のパラメータを利用した．

5.2 学習データの標準化を適用した実験

まず，従来手法と，従来手法における報酬推定 NN の更新に学習データの標準化を適用した手法を比較した．エージェントに対して，グローバル報酬，差分報酬，従来手法により近似した差分報酬，または学習データの標準化を適用して近似差分報酬を与えた場合の，平均獲得報酬の推移を**図 4** に示す．従来手法により近似した差分報酬を与えた場合は，序盤から中盤はグローバル報酬を用いた場合と変わらない報酬を獲得し，終盤にグローバル報酬を与えた場合と差分報酬を与えた場合の間の報酬を獲得した．一方，従来手法に学習データの標準化を適用した場合は，序盤ではグローバル報酬を与えた場合と差分報酬を与えた場合の間の報酬を獲得したが，中盤以降では差分報酬を与えた場合と変わらない報酬を獲得した．

また，問題ごとの獲得報酬を最終エピソードにおいて比較した結果を**表 1** に示す．ほとんどの問題で，学習データの標準化を適用した場合は従来手法以上の報酬を獲得した．一方，問題 3 や問題 5 のように，従来手法よりも低い報酬しか獲得できなかった問題もあった．しかし，従来手法より獲得報酬が低かった問題において，他の報酬を与えた場合の結果も確認すると，差分報酬を与えた場合の獲得報酬がグローバル報酬や従来手法により近似した差分報酬を与えた場合より低くなっていた．そのため，このような問題においては，標準化を適用したことにより差分報酬の獲得報酬に近づいた．

5.3 ミニバッチ勾配降下法を適用した実験

次に，ミニバッチ勾配降下法を適用することによる影響を確認した．学習データの標準化のみを適用した場合と，学習データの標準化に加えミニバッチ勾配降下法を適用した場合の平均獲得報酬の推移を**図 5** に示す．学習開始から 500 エピソード付近までは，標準化のみを適用した手法と標準化に加えミニバッチ勾配降下法を適用した手法の結果にあまり差はないが，500 エピソードから 1000 エピソードの間では，標準化に加えミニバッチ勾配降下法を適用した場合の方が獲得報酬の上昇が速かった．中盤以降では，標準化に加えミニバッチ勾配降下法を適用した手法は，差分報酬や標準化のみを適用した手法と変わらない報酬を獲得した．

5.4 考察

5.2 節の図 4 で示した結果から，学習データの標準化を適用した手法は従来手法に比べ平均獲得報酬が高いため有用であるといえる．また，問題ごとの獲得報酬を比較した表 1 では，学習データの標準化を適用することにより従来手法より獲得報酬が低くなった問題もあったが，そのような問題では差分報酬が高い報酬を獲得できていなかったため，従来手法より差分報酬を用いた場合の結果に近づいたと考えられる．いずれの結果からも，従来手法に学習データの標準化を適用することにより，差分報酬の結果に近づき，従来手法に比べ結果が改善されたと考えられる．

5.3 節の図 5 で示した結果から，学習データの標準化に加えミニバッチ勾配降下法を適用した手法は，学習データの標準化のみを適用した手法に比べ獲得報酬の上昇が速かったことがわかる．差分報酬を用いた場合の獲得報酬の

上昇が最も速かったことを考えると，ミニバッチ勾配降下法を適用することにより，学習データの標準化のみを適用した場合よりもさらに差分報酬を用いた場合の結果に近づいたといえる．

本研究ではエージェント数と POI 数が比較的少ない環境における提案手法の有効性を示したが，規模を拡大した環境においては改善の余地があり，適用するニューラルネットワークおよび，部分観測性を持つエージェントの状態空間の扱いを含めた検討が今後の課題である．

6. まとめ

本研究では，各エージェントが差分報酬を直接的に計算することが困難なマルチエージェント強化学習における，近似差分報酬の学習手法の改善を目的として，学習データの標準化とミニバッチ勾配降下法を適用した．比較的小さなベンチマーク問題の環境において，2 つの改良手法を推定報酬の学習に適用することにより，マルチエージェント強化学習系全体の獲得報酬が改善されることを示した．今後の課題は，ニューラルネットワークに適用する手法やエージェントの部分観測性について検討し，規模を拡大した環境において提案手法の有効性を示すことである．

参考文献

[1] Cao, Y., Yu, W., Ren, W. and Chen, G.: An overview of recent progress in the study of distributed multi-agent coordination, *IEEE Transactions on Industrial informatics*, Vol. 9, No. 1, pp. 427–438 (2012).
[2] Van der Pol, E. and Oliehoek, F. A.: Coordinated deep reinforcement learners for traffic light control, *Proceedings of Learning, Inference and Control of Multi-Agent Systems (at NIPS 2016)* (2016).
[3] Busoniu, L., Babuska, R. and De Schutter, B.: 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).
[4] Castellini, J., Devlin, S., Oliehoek, F. A. and Savani, R.: Difference Rewards Policy Gradients, *arXiv preprint arXiv:2012.11258* (2020).
[5] Colby, M., Duchow-Pressley, T., Chung, J. J. and Tumer, K.: Local approximation of difference evaluation functions, *Proceedings of the 2016 International Conference on Autonomous Agents & Multiagent Systems*, pp. 521–529 (2016).
[6] Parker, L. E., Rus, D. and Sukhatme, G. S.: Multiple mobile robot systems, *Springer Handbook of Robotics*, Springer, pp. 1335–1384 (2016).
[7] Tumer, K. and Agogino, A.: Distributed agent-based air traffic flow management, *Proceedings of the 6th international joint conference on Autonomous agents and multiagent systems*, pp. 1–8 (2007).
[8] Malialis, K., Devlin, S. and Kudenko, D.: Resource abstraction for reinforcement learning in multiagent congestion problems, *arXiv preprint arXiv:1903.05431* (2019).
[9] Colby, M. and Tumer, K.: Shaping fitness functions for coevolving cooperative multiagent systems, *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems-Volume 1*, pp. 425–432 (2012).
[10] Agogino, A. and Tumer, K.: Efficient Evaluation Functions for Evolving Coordination.
[11] Josh, P. and Adam, G.: Deep learning: A practitioner's approach，O'Reilly Media, Inc. (2017).
[12] Ruder, S.: An overview of gradient descent optimization algorithms, *arXiv preprint arXiv:1609.04747* (2016).
[13] Goodfellow, I., Bengio, Y. and Courville, A.: *Deep learning*, MIT press (2016).
[14] Zerbel, N.: zerbeln/S-DPP, , available from ⟨https://github.com/zerbeln/S-DPP⟩ (accessed 2021-09-11).