CF-002

見間違えのある繰り返し囚人のジレンマにおける方策勾配法に関する研究
Policy Gradients in Prisoner's Dilemma with Noisy Observations

坂本 充生 * 阿部　拳之 † 岩崎 敦 *
Mitsuki Sakamoto Kenshi Abe Atsushi Iwasaki

概要

本論文は見間違えのある繰り返し囚人のジレンマにおいて，均衡戦略に相当する方策を学習する手法を提案し，従来の方策勾配法などと比較，評価する．人がどのようにして協力するのかは人工知能や経済学，生物学における基本的な問題である．一般には，見間違えのある繰り返し囚人のジレンマの均衡計算は非常に難しい問題であることが知られている．実際，見間違えの下では，従来よく知られているしっぺ返し戦略でも，どちらかがいったん裏切ると，協力状態に戻るのは難しくなる．そこで本研究では，試行錯誤から報酬の高い方策を学習する強化学習的な手法が厳密な均衡ソルバの代わりに機能するかを吟味する．そこで，本論文では突然変異付きレプリケータダイナミクスの仕組みを利用した Neural Replicator-Mutator Dynamics (NeuRMD) を提案する．NeuRMD は方策勾配法と似た方法で導出できるシンプルな学習アルゴリズムである．計算実験の結果，シンプルだが非自明な均衡戦略である Win-Stay，Lose-Shift に相当する方策を安定的に学習することがわかった．

1　はじめに

人がどう協力する／しないかの仕組みは人工知能，経済学，生物学などにまたがった学際的な研究課題である．とくに繰り返しゲームは長期的関係にあるプレイヤ間の（暗黙の）協調を説明するためのモデルであり [10]，主に経済学分野で企業間の談合といった協調行動を分析するために発展してきた [23]．暗黙の協調を実現するには，プレイヤが相手の行動をある程度観測できることが前提となる．これまで，相手の行動が完全に観測できる完全観測（perfect monitoring）のケースについては多く論じられている [2, 12, 14]．

しかし，現実には相手の行動が完全に観測できない不完全観測（imperfect monitoring）のケース，つまり，プレイヤが相手の行動を見間違える場合がある [8, 22, 24]．これの特徴は，プレイヤが相手の行動に関してノイズを含む観測（シグナル）を私的に受け取ると仮定する点にある．いいかえると，あるプレイヤが相手の行動について観測したシグナルと異なるシグナルを他のプレイヤが観測しているかもしれないと仮定している．不完全観測付き無限回繰り返しゲームおいてどのような振る舞い (戦略) が最適なのかについては，ゲーム理論における代表的なゲームである囚人のジレンマの例でさえ十分にわかっていない．例えば，部分観測可能マルコフ決定過程 (Partially Observable Markov Decision Process, POMDP) を用いて均衡を計算する手法 [21] が知られているが，その計算量は一般には決定不能とされている．

POMDP に比べて高速に計算できる手法として，試行錯誤から報酬の高い方策を学習する強化学習がある．しかし，マルチエージェント系における，その帰結は収束するか否かや収束するとしても均衡に到達するかどうかはよくわかっていないことが多い．また，そのほとんどが完全観測のケースであり，常に裏切るような戦略しか学習できないことが多い [17, 4]．もう１つの手法として，レプリケータダイナミクス [16, 7] がある．これは，進化ゲーム理論でよく用いられるダイナミクスの１つであり，頻度依存淘汰モデルを用いて最適な戦略を探る．この手法では，利得が高くなる戦略をとるプレイヤの人口は増加し，低くなる戦略をとる人口はより良い戦略へ取って代わられてやがて絶滅するといった具合に自然淘汰の過程を表現する．このダイナミクスを計算するには，事前に戦略を列挙する必要がある．戦略空間を有限状態機械に限定したレプリケータダイナミクスの下では，見間違えが起きても協力状態を回復しやすい戦略が生き残ることが明らかになっているが，より複雑な戦略を分析することは困難である [26]．

そこで本研究では，まず繰り返し囚人のジレンマをプレイするエージェントを実装するために，エージェント自身の行動と観測したシグナルの履歴を状態としたマルチエージェント系の学習アルゴリズムを表現する枠組みを定義する．既存の学習アルゴリズムとして，まず方策勾配法の１つである q-based Policy Gradient (QPG) [15] を取り上げる．次に，QPG とよく似た方法で，突然変異のないレプリケータダイナミクスから各行動への重み（ロジット）の更新式を導出した Neural Replicator Dynamics (NeuRD) [6] を扱う．QPG は相手が振る舞いを戦略的に変えるような非定常的な状況では学習がうまくいかないことが知られている．NeuRD はこれを改良し，二

* 電気通信大学大学院情報理工学研究科
† 株式会社サイバーエージェント

人零和不完全情報展開型ゲームにおいてナッシュ均衡への収束を保証することに成功している．

本研究では零和ゲームから囚人のジレンマのようなゲームでも均衡に近い方策を学習できるよう突然変異付きレプリケータダイナミクスから学習アルゴリズムである Neural Replicator-Mutator Dynamics (NeuRMD) を提案する．見間違えのある戦略的環境下で，3 つの手法のそれぞれでエージェントがどんな振る舞いを学習するか吟味した．その結果，NeuRMD がもっとも高い利得を実現する方策を安定的に学習することを明らかにした．この方策は経済学や生物学で Win-Stay, Lose-Shift (WSLS) という戦略に相当する [9, 13]．実際，WSLS は見間違えのある環境で均衡となり最も高い利得を実現する戦略として知られている [21, 26].

2 モデル

本章では文献 [21] に基づいて, 2 人私的観測付き無限回繰り返しゲームをモデル化する．ここでプレイヤ $i \in \{1, 2\}$ は成分ゲームを無限期間 $t = 0, 1, 2, \cdots$ に渡って繰り返す. 各期においてプレイヤ i は有限集合 A から行動 a_i を選択し, その行動の組を $\boldsymbol{a} = (a_1, a_2) \in A^2$ とする. 次に, プレイヤ i は $\boldsymbol{a}$ に関する私的なシグナル $\omega_i \in \Omega$ を観測する. $\boldsymbol{w}$ をシグナルの組 $(\omega_1, \omega_2) \in \Omega^2$ とする. また, プレイヤが $\boldsymbol{a}$ を選択したとき $\boldsymbol{w}$ が生起する同時確率を $o(\boldsymbol{w} \mid \boldsymbol{a})$ とし, この同時確率を与える分布のことをシグナル分布と呼ぶ. 成分ゲームは無限回繰り返し行われるので，プレイヤ i の割引利得和は割引因子 $\delta \in (0, 1)$ により $\sum_{t=1}^{\infty} \delta^t g_i(\boldsymbol{a}^t)$ となる. ただし，$g_i(\cdot)$ の値は利得表によって定められた値に従う．

このとき，有限集合 Ω に対する $o_i(\omega_i \mid \boldsymbol{a})$ を Ω の限界分布 (marginal distribution) とする．加えて，どのプレイヤも他のプレイヤが選択した (または選択しなかった) 行動を正確には分からないと仮定する．どの行動プロファイル $\boldsymbol{a}$ に対しても，それぞれのシグナルプロファイル $\boldsymbol{\omega}$ が生起する確率は正となる．プレイヤ i の行動 a_i とシグナル ω_i から “実現利得 (realized payoff)” $\pi_i(a_i, \omega_i)$ を

$$g_i(\boldsymbol{a}) = \sum_{\boldsymbol{w} \in \Omega^2} \pi_i(a_i, \omega_i) o(\boldsymbol{w} \mid \boldsymbol{a})$$

を満たすように選ぶ.

表 1 に囚人のジレンマの利得表を示す．表中の C は協力行為を，D は裏切り行為を表す．囚人のジレンマの利得構造は $g > 0,\ l > 0$ であり，このとき D は厳密な支配戦略となる．また、囚人にジレンマでは $|g - l| < 1$ が要求される．もしこの条件が成り立たないとすると，繰り返し囚人のジレンマにおいて協力と裏切りを交互に出すほうが，純粋な協力よりも利得が高くなってしまい，純粋な協力が維持できなくなる．

次にプレイヤ 2 の行動に関するプレイヤ 1 のノイズを含む観測をプレイヤ 1 の私的シグナルとし, $\omega \in \{g, b\}$ (*good*, *bad*) とする．正しい観測ではプレイヤ 2 が C を選択した際のプレイヤ 1 の私的シグナルは g, D を選択した際の私的シグナルは

表 1: 囚人のジレンマ ($g > 0,\ l > 0,$ および $|g - l| < 1$)

	$a_2 = C$	$a_2 = D$
$a_1 = C$	$1, 1$	$-l, 1 + g$
$a_1 = D$	$1 + g, -l$	$0, 0$

表 2: (C, C) のときのシグナル分布

	$w_2 = g$	$w_2 = b$
$w_1 = g$	p	q
$w_1 = b$	q	$1 - p - 2q$

b となる. プレイヤ 2 についても同様である． よく使われる不完全私的観測のシグナル分布にほぼ完全観測がある． ここでは，両プレイヤが正しいシグナルを観測する確率は p, 片方のプレイヤが間違ったシグナルを観測する確率はそれぞれ q とする．また，$1 - p - 2q$ の確率で両方のプレイヤが間違ったシグナルを観測する．例として，(C, C) が実現した場合のシグナル分布を表 2 に示す． ただし，両プレイヤが正しいシグナルを観測する確率 p が最も高くなるように設定する．

繰り返し囚人のジレンマの均衡戦略として，“勝ち残り，負け逃げ” (Win-Stay, Lose-Shift, WSLS) [13] がある (図 1). WSLS は状態 R からスタートし，相手の協力を観測したときは同じ状態に留まり，裏切りを観測するともう一つの状態へと遷移する．裏切りを観測してから協力に戻るのは一見不自然に見えるが，お互いを処罰してから協力に戻ることで，見間違えのある環境で有名なしっぺ返し戦略より協力状態を維持しやすくなっている．本論文で提案するアルゴリズムは状態 P からスタートする WSLS を安定して学習することに成功した．

数ある戦略の中から有効な戦略を発見する方法の 1 つとして，レプリケータダイナミクスがある．ゲームを行うプレイヤの集団を考え，プレイヤはいくつかの戦略の中からランダムに戦略を選択し，他のプレイヤとゲームを行い利得を得る．その後, 戦略の集団に対する利得と集団全体の平均利得との差に応じて戦略の人口比を増減させる [14]．本論文ではレプリケータダイナミクスに突然変異の概念を加える．この突然変異付きレプリケータダイナミクスでは，適応度による人口の変化に加えて，すべての戦略が適応度に関係なく一定の確率で突然変異し異なる戦略をとるようにする．ある戦略が突然変異する確率を u とした突然変異付きレプリケータ方程式が

$$\dot{x_i} = x_i \left[f_i(\overrightarrow{x}) - \phi(\overrightarrow{x}) \right] + u \left(\frac{1}{n} - x_i \right),\ i = 1, ..., n \quad (1)$$

となる [20]. ここで $\phi(\cdot)$ を全ての戦略の利得の平均 $\sum_j x_j f_j(\overrightarrow{x})$, $f_j(\cdot)$ を $\sum_m x_j a_{jm}$ とする．ただし，a_{jm} は戦略 j をとるプレイヤが戦略 m を取るプレイヤと無限回プレイしたときの割引利得和である．

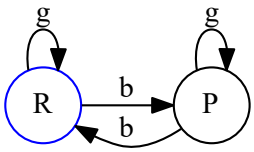

図 1: WSLS

3 見間違えのある環境とマルチエージェント強化学習

強化学習は，エージェントと呼ばれる意思決定主体が適切な振舞いを学習するアルゴリズムである．強化学習では，ある時刻 t において，エージェントは環境から状態 $s_t \in \mathcal{S}$ を観測する．次に行動 $a_t \in \mathcal{A}$ を自身の方策 $\boldsymbol{\pi} : s \to \Delta\mathcal{A}$ に従って決定する．エージェントは行動に応じて，環境から報酬 $r_t \in \mathbb{R}$ と新たな状態 s_{t+1} を受け取る．強化学習の目的は，状態 s における期待割引累積報酬 $\mathbb{E}[\sum_{k=t}^{\infty} \delta^{k-t} r_k | s_t = s]$ を最大化するような方策を学習することである．よく知られている強化学習手法，Q 学習 [18] は，今日の状態と行動によって報酬と次の状態遷移が決定されるモデル，マルコフ決定過程 (Markov decision process; MDP) において最適な方策の学習が保証されている [19].

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方策を学習するため，シングルエージェントの学習と比較して様々な問題が存在する．マルチエージェント強化学習では，時刻 t において複数のエージェントがそれぞれ状態 $s_{t,i}$ を観測し，行動 $a_{t,i}$ を決定する．そのため，エージェント i が受け取る報酬 $r_{t,i}$ と次の状態 $s_{t+1,i}$ は，自分の行動 $a_{t,i}$ だけでなくその他のエージェントの行動 $a_{t,-i}$ の行動によって決定される．また他のエージェントも同時に方策を学習するため，振舞いが非定常的に変化する．このような非定常性や不確実性にどう対応するかがマルチエージェント強化学習の重要なトピックである．

本研究では，見間違えのある繰り返し囚人のジレンマ上に，マルチエージェント強化学習を定義する．繰り返しゲームで，エージェントは過去の相手の行動から自身の振る舞いを決定する．また見間違えのある環境において，エージェント i が観測できるのは，ある時刻 t の自分の行動 $a_{t,i}$ と環境から観測するシグナル $\omega_{t,i}$ である．そのため，エージェント i が観測する状態 s_t は時刻 t における任意の過去 n 期前までの履歴 $s_{t,i} = (a_{k,i}, \omega_{k,i})_{k=t-n}^{t-1}$ とする．ただし，本研究では状態を，過去 1 期の履歴に限定する．つまり，エージェントに与える状態集合を

$$\mathcal{S} = \{\emptyset, Cg, Cb, Dg, Db\}$$

とする．ここで $\emptyset$ はゲームの始まりにおける状態を指す．またエージェントには，報酬から相手の行動を推定させないため，報酬 r には実現利得 $r(a, \omega)$ を与える．この時 $r(a, \omega)$ は利得構造とシグナル分布から

$$r(C, g) = \frac{p+q}{2p+2q-1} 1 + \frac{1-p-q}{2p+2q-1} l,$$
$$r(C, b) = -\frac{p+q}{2p+2q-1} l - \frac{1-p-q}{2p+2q-1} 1,$$
$$r(D, g) = \frac{p+q}{2p+2q-1} (1+g), \text{ and}$$
$$r(D, b) = -\frac{1-p-q}{2p+2q-1} (1+g)$$

となる．無限回繰り返しゲームでは，割引因子 δ が定める確率で次もゲームが継続するかを定める．ここで，一度ステージゲームをプレイすることを 1 step，1 つの繰り返しゲームが終了するまでを 1 episode とする．マルチエージェント強化学習はこの episode を繰り返すことで学習をする．

3.1 方策勾配法

方策勾配法は，方策勾配を用いて獲得報酬を最大化するような方策を求める強化学習の一手法である [25]．まず，強化学習の方策 $\boldsymbol{\pi}_\theta = (\pi(a|s;\theta))_{s \in S, a \in A}$ がパラメータ $\boldsymbol{\theta}$ によって決まる関数であると考える．またこの方策 $\boldsymbol{\pi}_{\boldsymbol{\theta}}$ に従った場合，時刻 T[*1] までの行動履歴

$$\tau_{\boldsymbol{\pi}_{\boldsymbol{\theta}}} = (s_0, a_0, s_1, a_1, ..., s_T, a_T, s_{T+1})$$

が与える期待累積報酬 $R(\tau_{\boldsymbol{\pi}_{\boldsymbol{\theta}}}) = \sum_{t=0}^{T} r(s_t, a_t)$ が決まる．これを $\boldsymbol{\theta}$ に関する関数

$$J(\boldsymbol{\theta}) = \mathbb{E}_{\boldsymbol{\pi}_{\boldsymbol{\theta}}}[R(\tau)] = \mathbb{E}_{\boldsymbol{\pi}_{\boldsymbol{\theta}}}[\sum_{t=0}^{T} r(s_t, a_t)]$$

として，$J(\boldsymbol{\theta})$ を最大化するような $\boldsymbol{\theta}$ を求めることが方策勾配法の目的である．この $\boldsymbol{\theta}$ を求めるために，方策勾配

$$\nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta}) = \mathbb{E}_{\boldsymbol{\pi}_{\boldsymbol{\theta}}}[\nabla_{\boldsymbol{\theta}} \log(\pi_{\boldsymbol{\theta}}(a|s)) Q^{\boldsymbol{\pi}_{\boldsymbol{\theta}}}(s, a)]$$
$$s.t.\ Q^{\boldsymbol{\pi}_{\boldsymbol{\theta}}}(s, a) = E_{\boldsymbol{\pi}_{\boldsymbol{\theta}}}[\sum_{k=t}^{\infty} \delta^{k-t} r_k | s_t = s], \forall s$$

を用いて，

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t + \eta \nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta}) \tag{2}$$

と更新する．ここで，η は学習率を示す．このように方策勾配法では，$\boldsymbol{\theta}$ を更新することで，適切な方策を学習する．

また本研究では，Mean actor critic (MAC) を用いて学習を行った．Actor-Critic 法は，方策をもつ Actor と方策評価をする Critic から構成され，Critic が現在の方策を評価し，その評価から Actor が方策を更新する．通常の Actor-Critic 法では，$\nabla_\theta J(\theta)$ を推定するために，サンプルした状態と行動を使用してモンテカルロ近似をしていたが，MAC はサンプルした状態のみを使ってモンテカルロ近似をする [1]．具体的には，従来の Actor-Critic 法は，Critic として状態価値 V 値 $V^{\boldsymbol{\pi}}(s)$ を用いて方策評価をし，サンプルした状態と行動

[*1] 一般的にこの T は 1 *episode* あたりのステップ数であり，方策はエピソードごとに更新される．

のみの勾配から方策を更新する．MAC では状態行動価値である Q 値を用いて方策評価し，サンプルした状態のすべての行動についての勾配から方策を更新する．ここで，状態価値 $V^{\boldsymbol{\pi}}(s) = \sum_a \pi(a|s) Q^{\boldsymbol{\pi}}(s,a)$ と計算する．また方策の評価には Q 値の代わりに，アドバンテージ

$$A^{\boldsymbol{\pi}}(s,a) = Q^{\boldsymbol{\pi}}(s,a) - V^{\boldsymbol{\pi}}(s)$$

を用いる．

MAC は別名 **q-based Policy Gradient (QPG)** とも呼ばれる [15]．QPG では式 2 から $\boldsymbol{\theta}$ を以下のように更新する：

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t + \eta \mathbb{E}_{s\sim d^{\boldsymbol{\pi}_{\boldsymbol{\theta}}}(\cdot)}[\sum_a \nabla_{\boldsymbol{\theta}} \pi(a|s;\boldsymbol{\theta}) A^{\boldsymbol{\pi}_{\boldsymbol{\theta}}}(s,a)] \quad (3)$$

ここで，$d^{\boldsymbol{\pi}}(s) = \mathbb{E}_{\boldsymbol{\pi}}[\sum_{t=0}^{\infty} \delta^t P(s_t = s|s_0, \boldsymbol{\pi})]$ は期待割引累積訪問数である．方策勾配法では様々な方策を表現することができるが，本研究では行動空間が離散であるため，特にロジット $\boldsymbol{y} = (y(s,a;\theta))_{s\in S, a\in A}$ から，ソフトマックス関数

$$\pi(a|s;\boldsymbol{\theta}) = \frac{\exp\left(y(s,a;\boldsymbol{\theta})\right)}{\sum_{a'} \exp\left(y(s,a';\boldsymbol{\theta})\right)} \quad (4)$$

を用いて方策を導出する．これは以下の 2 つの手法でも同様である．また本研究では，$\boldsymbol{\theta}$ をニューラルネットワークなどの関数パラメータではなく，ロジットが $\boldsymbol{y} = (\theta(s,a))_{s\in\mathcal{S}, a\in\mathcal{A}}$ のように表形式 (Tabular Case) で表現される場合を考える．Tabular Case では QPG の更新式 3 を，ロジット $\boldsymbol{y}$ を次のように更新する:

$$y_{t+1}(s,a) = y_t(s,a) + d^{\boldsymbol{\pi}}(s) \eta \pi(a|s) A^{\boldsymbol{\pi}}(s,a) \quad (5)$$

QPG ではロジット $\boldsymbol{y}$ の更新を式 5 にしたがって Algorithm 1 を構成する．まずロジット $\boldsymbol{y}$ と方策の評価値である Critic の Q 値 $\boldsymbol{Q}^{\boldsymbol{\pi}}$ を初期化する．次に以下の処理を指定されたエピソードの数だけ繰り返す．まずは前の t エピソード目に更新したロジット $\boldsymbol{y}_t$ から，方策 $\boldsymbol{\pi}_t$ をソフトマックス関数を用いて計算する．次にエージェントはこの方策 $\boldsymbol{\pi}_t$ に従って，エピソードが終了するまで，状態 s_n に対して行動 a_n を選択し，その結果報酬 r_n と次の状態 s_{i+n} を受け取る．この状態と行動，報酬の履歴に従って，現在の方策 $\boldsymbol{\pi}_t$ を評価するために，Q 値 $\boldsymbol{Q}^{\boldsymbol{\pi}_t}$ を更新する．そして，この Q 値 $\boldsymbol{Q}^{\boldsymbol{\pi}}$ から，そのエピソードで到達したすべての状態 s とすべての行動 a について，アドバンテージ $A^{\boldsymbol{\pi}}(s,a)$ を計算し，ロジット $y_{t+1}(s,a)$ を更新する．ここで，$d^{\boldsymbol{\pi}}(s)$ はその状態 s を訪問した回数を履歴の長さ n で割ることで，モンテカルロ近似する．

次に，**Neural Replicator Dynamics (NeuRD)** [6] を概説する．これは MAC 法を用いた強化学習手法の 1 つであり，レプリケータダイナミクスに基づいて，マルチエージェント系のような非定常な環境において性能を発揮する．具体的には，二人零和不完全情報展開型ゲームにおいてナッシュ均衡への収束が保証される．レプリケータダイナミクスは突然変異付きレプリケータダイナミクスの突然変異が $u = 0$ の特殊なケースであり，微分方程式

$$\dot{x_i} = x_i \left[f_i(\overrightarrow{x}) - \phi(\overrightarrow{x})\right]$$

Algorithm 1 QPG: Tabular Case, Mean Actor-Critic

Initialize logits $\boldsymbol{y_0}$ and critic Q-Vaule $\boldsymbol{Q}^{\boldsymbol{\pi}}$
for $t = 0$ to $number\_of\_episodes$ **do**
  $\boldsymbol{\pi}_t = softmax(\boldsymbol{y}_t)$
  $n = 0$
  repeat
    Perform a_n according to policy $\pi_t(a_n|s_n)$
    Receive reward r_n and new state s_{n+1}
    $n \leftarrow n + 1$
  until episode is not finished
  for $i \in \{0, ..., n-1\}$ **do**
    $Q^{\boldsymbol{\pi}_t}(s_i,a_i) \leftarrow (1-\alpha) Q^{\boldsymbol{\pi}_t}(s_i,a_i) + \alpha\{r_i + \delta \sum_{a'} Q^{\boldsymbol{\pi}_t}(s_{i+1},a') \pi_t(a'|s_{i+1})\}$
  end for
  for $s \in \mathcal{S}$ **do**
    for $a \in \mathcal{A}$ **do**
      $A(s,a)^{\boldsymbol{\pi}_t} = Q^{\boldsymbol{\pi}_t}(s,a) - \sum_{a'} Q^{\boldsymbol{\pi}_t}(s,a') \pi_t(a'|s)$
      $y_{t+1}(s,a) \leftarrow y_t(s,a) + \frac{1}{n}\eta \sum_{i=0}^{n-1} \mathbf{1}[s = s_i] \pi_t(a|s) A^{\boldsymbol{\pi}_t}(s,a)$
    end for
  end for
end for

にしたがって戦略分布を更新する．NeuRD は，このレプリケータ方程式に合わせてロジットの更新式 5 を修正する．詳しい導出は省略するが，第 4 章で示す導出とほぼ同じである．結果的に QPG と NeuRD はロジットの更新式 5 のみが異なるようになる．QPG はアドバンテージ関数 $A^{\boldsymbol{\pi}}(s,a)$ に $\pi(a|s)$ をかけていたのに対して，NeuRD はアドバンテージ関数 $A^{\boldsymbol{\pi}}(s,a)$ をそのまま使ってロジットを更新する．具体的には

$$y_{t+1}(s,a) = y_t(s,a) + d^{\boldsymbol{\pi}}(s) \eta A^{\boldsymbol{\pi}}(s,a) \quad (6)$$

となる．こうすることで，一度選択確率が低くなった行動もその他の行動と同じだけ $\boldsymbol{y}(s,a)$ を更新するようになる．この結果，マルチエージェント系のような非定常性の高い環境において，有効な方策が大きく変化した場合に適応できるようになる．

4 Neural Replicator-Mutator Dynamics

本節では **Neural Replicator-Mutator Dynamics (NeuRMD)** を新しく提案する．NeuRMD は突然変異付きレプリケータダイナミクス [20] から，NeuRD と同じようにロジットの更新式を導く．結果として NeuRD に突然変異

項を追加した形になる．

レプリケータダイナミクスに突然変異を導入するとそのダイナミクスの帰結が安定しやすくなることが知られている．突然変異がないレプリケータダイナミクスの平衡点が，同じ適応度をもつ連続した点になることが多く，1 つの平衡点を選ぶのが難しい．突然変異を導入することで，このような中立的に連続している平衡点を離散化することができる [5]．また，突然変異がない場合，漸近的に安定な平衡点の存在を保証できない．一方で突然変異があれば，その確率が 0 に収束するとき，漸近的に安定な平衡点の存在を保証できるようになる [3]．加えて，数値的なダイナミクスの計算も安定する．したがって，突然変異付きレプリケータダイナミクスを用いてロジット更新を構成することで，こうした利点を NeuRD に追加できる．

具体的な導出に備えて，まずレプリケータダイナミクスと方策勾配法の対応について考える．式 1 に示すレプリケータダイナミクスにおけるある戦略 i の人口 x_i はその戦略を取る確率に相当し，これは行動 a の選択確率 $\pi(a)$ に対応する．次に戦略 i が得る利得 $f_i(\vec{x})$ は状態行動価値 Q 値と，また全ての戦略の利得の平均である $\phi(\vec{x})$ は状態価値 V 値と対応する．また突然変異項に着目すると，n は戦略数であるため，行動数 $|\mathcal{A}|$ と対応する．以上を用いてレプリケータ方程式を方策の変化量の式に書き換える：

$$\begin{aligned}\dot{\pi}(a) &= \pi(a)[Q^{\boldsymbol{\pi}}(a) - V^{\boldsymbol{\pi}}] + u\left(\frac{1}{|\mathcal{A}|} - \pi(a|s)\right) \\ &= \pi(a)A^{\boldsymbol{\pi}}(a) + u\left(\frac{1}{|\mathcal{A}|} - \pi(a)\right).\end{aligned} \tag{7}$$

次にこの方策の更新量を示す式 7 から Tabular Case におけるロジットの更新量を導出する．式 4 と同様に $\pi(a)$ をロジット $\boldsymbol{y}$ に関するソフトマックス関数で表すと，

$$\pi(a) = \frac{\exp\left(y(a)\right)}{\sum_{a'}\exp\left(y(a')\right)}$$

となる．これを両辺を時刻 t で微分すると

$$\begin{aligned}\dot{\pi}(a) = \frac{d\pi(a)}{dt} &= \frac{\frac{d}{dt}\exp\left(y(a)\right)}{\sum_{a'}\exp\left(y(a')\right)} - \frac{\exp\left(y(a)\right)\frac{d}{dt}\sum_{a'}\exp\left(y(a')\right)}{\left(\sum_{a'}\exp\left(y(a')\right)\right)^2} \\ &= \frac{\exp\left(y(a)\right)\frac{d}{dt}y(a)}{\sum_{a'}\exp\left(y(a')\right)} - \frac{\exp\left(y(a)\right)\sum_{a'}\exp\left(y(a')\right)\frac{d}{dt}y(a')}{\left(\sum_{a'}\exp\left(y(a')\right)\right)^2} \\ &= \pi(a)\frac{d}{dt}y(a) - \pi(a)\sum_{a'}\pi(a')\frac{d}{dt}y(a') \\ &= \pi(a)\left[\frac{d}{dt}y(a) - \sum_{a'}\pi(a')\frac{d}{dt}y(a')\right].\end{aligned}$$

また式 7 を代入して

$$\frac{d}{dt}y(a) - \sum_{a'}\pi(a')\frac{d}{dt}y(a') = A^{\boldsymbol{\pi}}(a) + \frac{u}{\pi(a)}(\frac{1}{|\mathcal{A}|} - \pi(a))$$

となる．ただし，この方程式はあらゆる解を許容してしまうため，$\sum_{a'}\pi(a'|s)\frac{d}{dt}y(a) = 0$ と仮定して解を限定する．これにより，ロジット $\boldsymbol{y}$ の変化量は

$$\frac{d}{dt}y(a) = A^{\boldsymbol{\pi}}(a) + \frac{u}{\pi(a)}(\frac{1}{|\mathcal{A}|} - \pi(a)) \tag{8}$$

となる．この式 8 から，状態を持たない RD の $\pi(a)$ と状態を持つ強化学習の $\pi(a|s)$ を対応付けてオイラー離散化することでロジットの更新式

$$y_{t+1}(s,a) = y_t(s,a) + \eta\left\{A(s,a)^{\boldsymbol{\pi}} + \frac{u}{\pi(a|s)}(\frac{1}{|\mathcal{A}|} - \pi(a|s))\right\} \tag{9}$$

を得る．

NeuRMD ではロジット $\boldsymbol{y}$ の更新を式 9 にしたがって Algorithm 2 を構成する．これは基本的に Algorlithm 1 と同じだが，レプリケータダイナミクスに従って方策を学習するため，16 行目のロジット $y(s,a)$ はすべての状態 s に関して更新していることに注意したい．

Algorithm 2 NeuRMD: Tabular Case, Mean Actor-Critic

1: Initialize logits $\boldsymbol{y_0}$ and critic Q-Vaule $\boldsymbol{Q^{\pi}}$
2: **for** $t = 0$ to $number\_of\_episodes$ **do**
3: $\quad\boldsymbol{\pi}_t = softmax(\boldsymbol{y}_t)$
4: $\quad n = 0$
5: $\quad$**repeat**
6: $\quad\quad$Perform a_n according to policy $\pi_t(a_n|s_n)$
7: $\quad\quad$Receive reward r_n and new state s_{n+1}
8: $\quad\quad n \leftarrow n + 1$
9: $\quad$**until** episode is not finished
10: $\quad$**for** $i \in \{0, ..., n-1\}$ **do**
11: $\quad\quad Q^{\boldsymbol{\pi}_t}(s_i, a_i) \leftarrow (1-\alpha)Q^{\boldsymbol{\pi}_t}(s_i, a_i) + \alpha\left\{r_i + \delta\sum_{a'}Q^{\boldsymbol{\pi}_t}(s_{i+1}, a')\pi_t(a'|s_{i+1})\right\}$
12: $\quad$**end for**
13: $\quad$**for** $s \in \mathcal{S}$ **do**
14: $\quad\quad$**for** $a \in \mathcal{A}$ **do**
15: $\quad\quad\quad A(s,a)^{\boldsymbol{\pi}_t} = Q^{\boldsymbol{\pi}_t}(s,a) - \sum_{a'}Q^{\boldsymbol{\pi}_t}(s,a')\pi_t(a'|s)$
16: $\quad\quad\quad y_{t+1}(s,a) \leftarrow y_t(s,a) + \eta\left\{A^{\boldsymbol{\pi}_t}(s,a) + \frac{u}{\pi(a|s)}(\frac{1}{|\mathcal{A}|} - \pi_t(a|s))\right\}$
17: $\quad\quad$**end for**
18: $\quad$**end for**
19: **end for**

5 計算機実験

5.1 実験設定

本節では QPG, NeuRD, NeuRMD の 3 つの手法が，見間違えのある繰り返し囚人ジレンマで獲得する報酬と相互協力率を確認する．この時，2 体のエージェントがそれぞれ同じ手法で独立して学習すると仮定する．各エージェントは行動 $a \in \{C, D\}$ を選択し，シグナル $\omega \in \{g, b\}$ を観測し，自身の

行動とシグナルに応じた報酬である実現利得 $r(a,\omega)$ を受け取る．囚人のジレンマの利得パラメータを $g=0.1, l=0.1$，割引因子 $\delta=0.9$ とする．この割引因子はエピソード内のゲームが継続する確率を表す．つまり確率 δ でゲームを継続し，確率 $1-\delta$ でそのエピソードを終了する．シグナル分布のパラメータは $p=0.95, q=0.01$，Q 値の学習率 $\alpha=0.1$, 方策の学習率 $\eta=0.02$, 割引率 $\gamma=0.9$ とする．また NeuRMD の突然変異確率は $\mu=0.01$ とする．割引因子 δ によるゲームの終了までを 1 エピソードとして，500,000 エピソード学習し，ランダムシードを変えながら行った 30 試行を評価する．

5.2 獲得報酬と相互協力率の推移

図 2 では，各手法が獲得した 2 体の平均報酬値の推移を表している．2 体の平均報酬値は，そのエピソードでエージェントが獲得した 2 体の平均合計報酬値と繰り返し回数の商とした．縦軸は 2 体の平均報酬値，横軸はエピソード数に対応し，区間数 10 の移動平均と信頼区間をプロットした．図 3 では，各手法ごとの相互協力率の推移を表している．相互協力率は，あるエピソードにおいて，お互いに行動 C をとった割合である．縦軸は相互協力率，横軸はエピソード数に対応し，区間数 10 の移動平均と信頼区間をプロットした．最終的な平均獲得報酬は，QPG が 0.014, NeuRD が 0.16，NeuRMD が 0.67 となった．また最終的な相互協力率は，QPG が 0.28×10^{-3}, NeuRD が 0.14，NeuRMD が 0.66 となった．このことからわかるように平均獲得報酬，相互協力率ともに提案手法がほかの 2 つの手法を上回った．

平均報酬値と相互協力率を比較すると，NeuRMD は高い報酬値を獲得していても，平均報酬値と相互協力率の差が小さい．エージェントが相互協力によって得る報酬値はそれぞれ 1 である．一方，2 体のエージェントのうち片方が裏切り D を選択した場合，$g=l$ であれば平均報酬値は 0.5 となり，両方 D を選択した場合，平均報酬値は 0 となる．このことから，NueRMD は，片方のエージェントによる搾取ではなく，双方のエージェントの協力行動によって高い報酬値を獲得していることがわかる．

5.3 獲得方策の評価

前節では NeuRMD が協力的な方策を安定的に学習することが明らかとなったが，本説ではその獲得した方策を分析する．図 4 は NeuRMD の獲得した方策の推移を表している．縦軸が行動の選択確率，横軸がエピソード数に対応し，各観測状態における行動 C の選択確率をプロットした．最終的に獲得した方策は $\pi(C|\emptyset)$ が 0.01，$\pi(C|Cg)$ が 0.99, $\pi(C|Cb)$ が 0.01, $\pi(C|Dg)$ が 0.01，$\pi(C|Db)$ が 0.99 であった．この方策では，シグナル g を観測した場合は昨日自分が取った行動と同じ行動を，シグナル b を観測した場合は昨日自分が取った行動と異なる行動を選択する．この方策はすでに図 1 に示した WSLS と初期状態だけが異なる [11]．つまり，状態 P からスタートする WSLS (Suspicious-WSLS, S-WSLS) が NeuRMD が獲得した方策に対応する．WSLS は見間違えのある環境で最も高い利得を実現する均衡戦略として知られている [21, 26]．以上より，NeuRMD は従来手法より安定的に，見間違えのある繰り返し囚人のジレンマの均衡戦略を求めることに成功した．

次に，方策とロジットの推移から突然変異の効果について分析する．図 5 は NeuRMD のロジットの推移を表している．縦軸がロジットの値，横軸がエピソード数に対応し，各状態，行動ごとにプロットした．18000 エピソードを境に，状態 Cg と Db における行動 C のロジットが急激に大きくなり，行動 D のロジットを逆転している．これは突然変異により今までと異なる行動を選んだことで，高い報酬を獲得した結果だと考えられる．このような逆転が起こる以前の方策やロジットの推移に見ても，一度選択確率が低くなった行動に対応するロジットの値は定期的に増加しており，それが高い報酬を獲得できるような状態になることで，S-WSLS のような協力的な方策を学習すると考えられる．

以上の実験は，ランダムシードを変えながら 30 試行の結果を示した．各試行で 3 つのアルゴリズムがどれだけ均衡に近い方策を獲得したのかを分析する．どのアルゴリズムも均衡戦略への収束を保証できない．さらに，見間違えのある繰り返し囚人のジレンマでは，相手の方策を固定したときの最適反応を計算することも非常に難しく求めた方策が均衡であることをチェックできない．そこで，求めた方策が均衡でないことを判定することを考える．つまり，相手が学習した方策を固定して，学習させた方策（反応方策）が，元の方策より高い報酬値を獲得すれば，元の方策は相手の方策に対する最適反応でない，と判定する．この判定を以下の手続きにしたがってすすめる．

1. 学習済みの方策 $(\boldsymbol{\pi}_1,\boldsymbol{\pi}_2)$ の組を考える．
2. それぞれの方策について反応方策 $\boldsymbol{\pi}_1^R$ と $\boldsymbol{\pi}_2^R$ を求める．反応方策は学習済みの方策を片方のエージェントで固定し，QPG で学習させることで求める．ここでエピソード数は 10000 回とした．
3. 次に，学習済みの方策 $(\boldsymbol{\pi}_1,\boldsymbol{\pi}_2)$ で獲得する各エージェントの報酬値 $(\boldsymbol{r}_1,\boldsymbol{r}_2)$ と，学習済みの方策と反応方策の組 $(\boldsymbol{\pi}_1,\boldsymbol{\pi}_1^R),(\boldsymbol{\pi}_2^R,\boldsymbol{\pi}_2)$ が獲得した報酬値 $(\boldsymbol{r}_{1,1},\boldsymbol{r}_{1,2}^R),(\boldsymbol{r}_{2,1}^R,\boldsymbol{r}_{2,2})$ を求める．
4. ここでの獲得報酬は各組について 10000 エピソードをプレイさせた結果の報酬値の平均とした．これは見間違いの発生や行動選択のモンテカルロ誤差を小さくするためである．
5. 最後に $\boldsymbol{r}_1$ と $\boldsymbol{r}_{2,1}^R$，$\boldsymbol{r}_2$ と $\boldsymbol{r}_{1,2}^R$ を比較することで最適反応性を評価する．具体的には，この学習済み方策の報酬値と反応方策の報酬値の差から，学習済み方策の最適反応性を定量的に評価する．

図 6 は各手法ごとの学習済み方策と反応方策の報酬値の差を示している．学習済み方策と反応方策の差は $r_i - r_{i,-i}^R$ とし，これが負になると反応方策で報酬を改善したことになり，元の学習済み方策が均衡を構成しないと判定できる．図 6 の縦軸は報酬差を示しており，すべてのランダムシードとエー

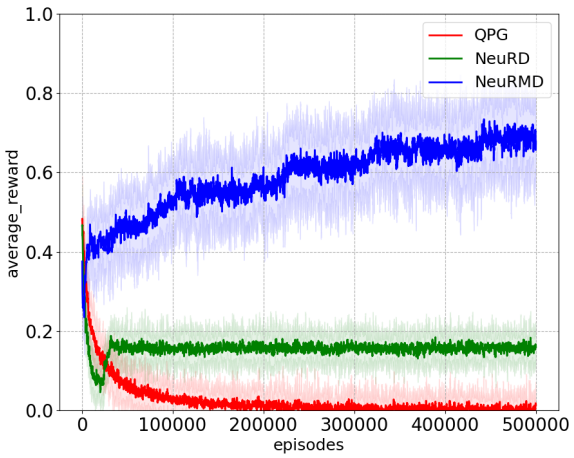

図 2: 2 体の平均獲得報酬の推移

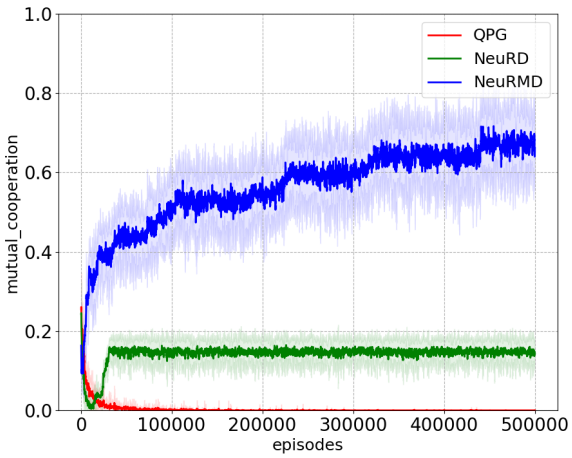

図 3: 相互協力率の推移

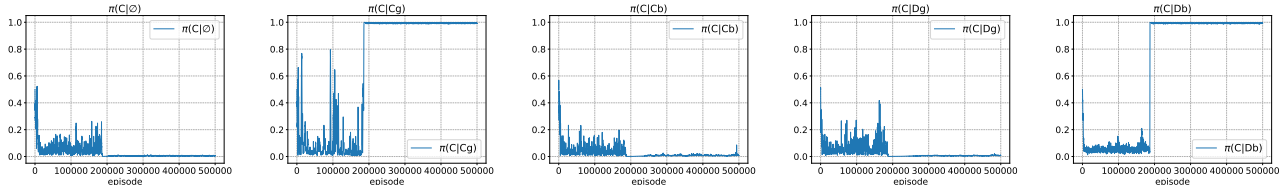

図 4: NeuRMD: 方策の推移

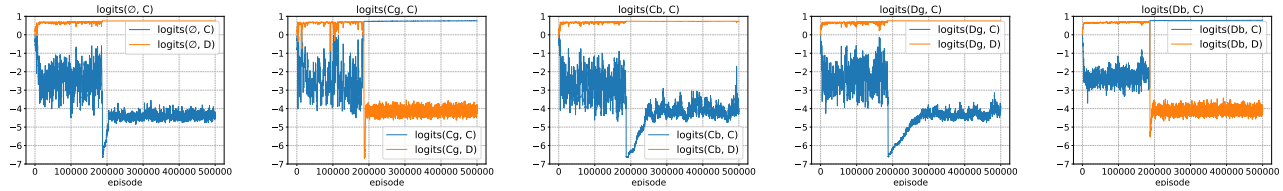

図 5: NeuRMD: ロジットの推移

ジェントに関して箱ひげ図を作成し，四分位からの外れ値をプロットした．その結果，報酬値の差の中央値はそれぞれ QPG が 0.46×10^{-3}，NeuRD が 0.40×10^{-2}，NeuRMD が 0.35×10^{-2} となり，NeuRD, NeuRMD, QPG の順に大きかった．QPG は元々 QPG 同士で学習しているため，報酬値の差が 0 に近いのは自然であると考えられる．次に，NeuRD は最も報酬値の差の中央値は大きいが，大きな外れ値が 60 個中 6 個あるため，均衡戦略を安定的に学習しているとは言えない．最後に，NeuRMD はその報酬値がすべて 0 以上であり，大きな外れ値もない．よって NeuRD より安定的に方策を獲得しているといえる．ここで，提案手法が従来手法である NeuRD よりも平均値や中央値で劣るのは，突然変異項による探索の影響であると考えられる．それでも，NeuRMD は協力的な均衡に近い戦略を安定的に獲得しているといえる．

6　おわりに

本論文は，見間違えのある繰り返し囚人のジレンマの均衡戦略に相当する方策を学習する手法として，進化ゲームでよく知られている突然変異付きレプリケータダイナミクスの仕組みを行動の重み更新に導入した NeuRMD を提案した．NeuRMD は非自明な協力的均衡戦略である WSLS を，従来手法より安定的に学習することを実験的に示した．一般に繰り返し囚人のジレンマで常に裏切る戦略以外を学習させることは難しいが，提案手法は，同じような構造を持つゲームにおいて，プレイヤが相手を処罰する機構をもちながら協力する仕組みを発見できることを示唆した．今後の課題として，2 期以上の記憶をもつケースの分析や，企業間の価格競争や企業の参入／退出といった行動空間を大きくしたゲームの分析が挙げられる．

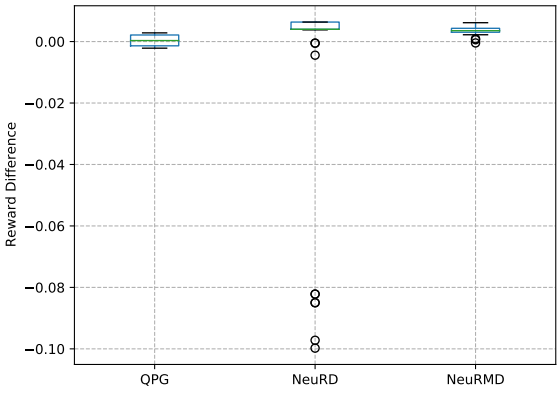

図 6: 反応方策との報酬差

参考文献

[1] C. Allen, K. Asadi, M. Roderick, A. rahman Mohamed, G. Konidaris, and M. Littman. Mean actor critic, 2018.

[2] R. Axelrod. *Genetic Algorithms and Simulated Annealing*, chapter The Evolution of Strategies in the Iterated Prisoner's Dilemma, pp. 32–41. 1987.

[3] J. Bauer, M. Broom, and E. Alonso. The stabilization of equilibria in evolutionary game dynamics through mutation: mutation limits in evolutionary games. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 475(2231):20190355, 2019.

[4] D. Bloembergen, K. Tuyls, D. Hennes, and M. Kaisers. Evolutionary dynamics of multi-agent learning: A survey. *J. Artif. Int. Res.*, 53(1):659–697, May 2015.

[5] I. M. Bomze and R. Burger. Stability by mutation in evolutionary games. *Games and Economic Behavior*, 11(2):146–172, 1995.

[6] D. Hennes, D. Morrill, S. Omidshafiei, R. Munos, J. Perolat, M. Lanctot, A. Gruslys, J.-B. Lespiau, P. Parmas, E. Duéñez-Guzmán, et al. Neural replicator dynamics: Multiagent learning via hedging policy gradients. In *Proceedings of the 19th International Conference on Autonomous Agents and MultiAgent Systems*, pp. 492–501, 2020.

[7] J. Hofbauer and K. Sigmund. *Evolutionary Games and Population Dynamics*. Cambridge University Press, Cambridge, 1998.

[8] M. Kandori. Repeated games. In S. N. Durlauf and L. E. Blume eds., *Game theory*, pp. 286–299. Palgrave Macmillan, 2010.

[9] D. Kraines and V. Kraines. Pavlov and the prisoner's dilemma. *Theory and Decision*, 26:47–79, 1989.

[10] G. Mailath and L. Samuelson. *Repeated Games and Reputation*. Oxford University Press, 2006.

[11] E. E.-S. Martin A. Nowak, Karl Sigmund. Automata, repeated games and noise. *Journal of Mathematical Biology*, 33:703–722, 1995.

[12] M. Nowak. *Evolutionary Dynamics: Exploring the Equations of Life*. Harvard University Press, 2006.

[13] M. Nowak and K. Sigmund. A strategy of win-stay, lose-shift that outperforms tit for tat in prisoner's dilemma. *Nature*, 364:56–58, 1993.

[14] K. Sigmund. *The Calculus of Selfishness*. Princeton University Press, 2010.

[15] S. Srinivasan, M. Lanctot, V. Zambaldi, J. Pérolat, K. Tuyls, R. Munos, and M. Bowling. Actor-critic policy optimization in partially observable multiagent environments. In *Proceedings of the Thirty-second Conference on Neural Information Processing Systems*, p. 3426–3439, 2018.

[16] P. D. Taylor and L. B. Jonker. Evolutionarily stable strategies and game dynamics. *Mathematical Biosciences*, pp. 145–156, 1978.

[17] K. Tuyls, Pieter Jan ' T, B. Vanschoenwinkel. An evolutionary dynamical analysis of multi-agent learning in iterated games. *Autonomous Agents and Multi-Agent Systems*, 12(1):115–153, 2006.

[18] C. Watkins. *Learning from delayed rewards*. PhD thesis, 1989.

[19] C. Watkins and P. Dayan. Q-learning. In *Machine Learning*, pp. 279–292, 1992.

[20] B. M. Zagorsky, J. G. Reiter, K. Chatterjee, and M. A. Nowak. Forgiver triumphs in alternating prisoner's dilemma. *PLOS ONE*, pp. 1–8, 2013.

[21] ヨンジュン, 岩崎, 神取, 小原, 横尾. 部分観測可能マルコフ決定過程を用いた私的観測付き繰り返しゲームにおける均衡分析プログラム. 情報処理学会論文誌, pp. 1234–1246, 2012.

[22] 関口. 経済セミナー増刊:ゲーム理論プラス, 「協調達成のための正しいお仕置きの仕方」. 日本評論社, 2007.

[23] 岡田. ゲーム理論 新版. 有斐閣, 2011.

[24] 松島. ゲーム理論の新展開. 勁草書房, 2002. 第 4 章:「繰り返しゲームの新展開:私的モニタリングによる暗黙の協調」, pp.89-114.

[25] 森村. 強化学習. 講談社, 2019.

[26] 西野上, 五十嵐, 岩崎. 私的観測下の繰り返し囚人のジレンマにおける協力のダイナミクス. 第 19 回情報科学技術フォーラム, 2020.