1L2-R-7-9

逆強化学習による複数均衡下での均衡収束の実現

Inverse Reinforcement learning to Play an Optimal Nash Equilibrium in Multiagent System

荒井 幸代[*1] 鈴木 香名子[*1] 大喜多 周[*2]
Sachiyo Arai Amane Okita Amane Okita

[*1]千葉大学大学院工学研究科 [*2]千葉大学工学部都市環境システム学科
Faculty of Engineering, Chiba University Department of Urban Environment Systems, Chiba University

The purpose of this study is to encourage a desirable actions among the multiple agents by means of estimating the reward function. We take two types of games, the coordination game and Stag Hunt game. Since both games have dual equilibria, the agent should be given an appropriate direction to converse into the same and optimal equilibrium

In this paper, we introduce *Inverse Reinforcement Learning (IRL)* which is required to be given an optimal sequence of actions to find the incentive in order to make agents reach global optimal solution. Our main contribution is to show how to apply IRL to design the reward under the environment of n-persons' game. Through some empirical results, we show the performances of mechanisms which were acquired by RL and IRL for solving dilemma of the coordination game and Stag Hunt game.

1. はじめに

本稿は，複数エージェントが局所的に相互作用を繰り返す環境を対象とし，エージェント間の相互作用は非協力ゲームで記述する．対象とするゲームには複数のナッシュ均衡が存在し，エージェント間で異なる均衡点への行動を選択すれば系全体のパフォーマンスが低下する場合がある．ある 1 つの均衡へと収束させる方法として，エージェント間の交渉や合意形成を図る仕組みの導入などエージェント間の直接的な通信を考える方法と，直接の合意形成なしに系全体が 1 つの均衡に収束するための適切な誘導方法に大別される．本稿は後者に属する．

具体的には，最適な行動系列から逆強化学習によって報酬関数を推定し，その報酬関数から求められる状態価値にしたがって行動するエージェントを環境に導入する．望ましい行動を所与とした時の行動設計法として紹介する．

2. 対象問題

表 1: Dual equilibria Games

(a) Coordination game

$i \setminus j$	L	R
L	$1,1$	$-1,-1$
R	$-1,-1$	$1,1$

(b) Stag-Hunt game

$i \setminus j$	Stag	Rabbit
Stag	100 ,100	0,10
Rabbit	10,0	10,10

本稿では，表 1(a)，(b) の利得行列で定義される協調ゲーム，および，スタグハントゲームを扱う．

協調ゲームのナッシュ均衡 (L,L) と (R,R) は，利得が等しく，行動選択における誘因やリスクが存在しないが，利得行列から最適な行動が特定できないという問題がある．このゲームタイプで L，R のいずれかの均衡に収束するためには，なんらかの誘導が必要となる．

一方，スタグハントゲームでは，$(\mathrm{Stag},\mathrm{Stag})$ がパレート最適であるが，相手が Rabit を選択したならば自分も Rabit を選択した方がよいことから，パレート最適が実現するとは限らない．このゲームタイプでもパレート優越解を実現するために両エージェントを共に Stag 選択に至らしめる適切な誘導が必要となる．

以下では，エージェントが強化学習の手法の 1 つである Q 学習 [Watkins 92] によって行動を獲得するとの前提で議論を進める．エージェントは状態 $s \in \mathcal{S}$ を知覚し，方策 π に基づいて行動 $a \in \mathcal{A}(s)$ を選択する．ただし，$\mathcal{S}$ は環境の遷移可能な状態の集合を，$\mathcal{A}(s)$ は状態 s において選択可能な行動の集合を表す．エージェントは行動選択後に報酬 r を受け取り，新しい状態 s' を知覚する．

Q 学習は状態 s と行動 a の価値 $Q(s,a)$ を式 (1) により更新する．ここで α $(0 < \alpha \leq 1)$ は学習率，γ $(0 \leq \gamma \leq 1)$ は割引率を表し，k は s において a を選択し，$Q(s,a)$ を更新した回数である．

$$Q_{k+1}(s,a) = Q_k(s,a) + \alpha\{r + \gamma \max_{a' \in \mathcal{A}(s')} Q_k(s',a') - Q_k(s,a)\} \quad (1)$$

Q 学習は期待報酬値を最大化する行動を獲得することから，ゴールで得られる報酬の多寡でなく，そこに至る確率に依存する．この確率は他エージェントに依存し，これらは各エージェントにとっては不確実である．そこで，唯一の望ましい均衡に収束させるためには，なんらかの誘導が必要である．以下の節では，各ゲームにおける Q 学習モデルを説明する．

2.1 問題 1:協調ゲーム (Coordination Game)

2.1.1 環境モデル

本稿は文献 [Sen 07][Mukherjee 08] のマルチエージェント系環境の設定に準ずる．

トーラス状の二次元格子上に配置された複数のエージェントが局所的に協調ゲームを繰り返す．各エージェントの対戦相手は近傍に位置するエージェント (以後，近傍エージェントと表記) の中から毎回ランダムに選ばれ，各エージェントはゲームで得た利得を報酬として学習を進める．この時，エージェントは対戦相手を区別することはできないが，対戦相手のとった行動は知覚できる．ここで近傍エージェントとは，各エージェントから距離 D 以内に位置するエージェントのことを指し，D はマンハッタン距離で表す．

連絡先: 荒井幸代，千葉大学大学院工学研究科，〒 263-8522，043-290-3316，sachiyo@faculty.chiba-u.jp

2.1.2 エージェントモデル

状態集合 $\mathcal{S}$ は表 2 に示す通り，過去 l ステップの自身の行動 a^i と相手の行動 a^j の組み合わせ $(a^i_{t-l}, a^j_{t-l}, a^i_{t-l+1}, a^j_{t-l+1}, \ldots, a^i_{t-1}, a^j_{t-1})$ とする．状態数は 2^{2l} となる．**行動**は L または R，**報酬**は表 1(a) に定義された利得を報酬 r として与える．

全エージェントが状態入力から行動を出力し，Q 値を更新するまでを 1 ステップとし，T ステップの繰り返しを 1 試行とする．1 ステップは以下の (1)-(3) である．n はエージェント数で，ここでは $n > 3$ である．

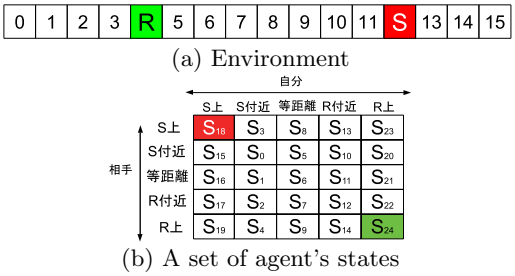
(1) 全エージェントは自身の方策に基づいて行動を選択する

(2) エージェント i $(= 1, 2, \ldots, n)$ は近傍エージェントの中から対戦相手 j $(\neq i)$ をランダムに選択する

(3) エージェント i はエージェント j とゲームを行い，獲得した利得 r_i に基づいて Q 値を更新する

表 2: $\mathcal{S}$: The set of states

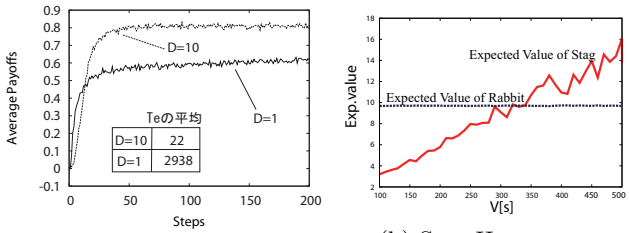
0	–			
1	s_1: LL	s_2: LR	s_3: RL	s_4: RR
2	s_1: LLLL	s_2: LLLR	s_3: LLRL	s_4: LLRR
	s_5: LRLL	s_6: LRLR	s_7: LRRL	s_8: LRRR
	s_9: RLLL	s_{10}: RLLR	s_{11}: RLRL	s_{12}: RLRR
	s_{13}: RRLL	s_{14}: RRLR	s_{15}: RRRL	s_{16}: RRRR
.	...			
l	$a^i_{t-l}a^j_{t-l}a^i_{t-l+1}a^j_{t-l+1}\cdots a^i_{t-1}a^j_{t-1}$			
.	.			



(a) Environment

(b) A set of agent's states

図 1: Multi-Step Stag-Hunt game



(a) Coordination game

(b) Stag-Hunt game

図 2: Learning curves of the agents

2.2 問題 2: スタグハントゲーム (Stag-Hunt Game

2.2.1 環境モデル：マルチステップ・スタグハントゲーム

本稿では, スタグハントゲームを一次元格子状のモデルに拡張し, その行動を観測する．通常のスタグハントゲームでは, 行動選択直後に報酬が得られるのに対し, 本拡張モデルは, 複数の行動系列後に報酬が得られることから, このゲームをマルチステップスタグハントゲームと呼ぶ．この拡張によって Stag, Rabbit に至る相手の時系列行動を観測して自分の行動を決める．

図 1(a) に拡張モデルの環境設定を示す．環境は 0〜15 番地の計 16 番地から成る．初期配置は，Rabbit が 4 番地，Stag が 12 番地, エージェント i,j はエピソードごとにランダムに配置する．

2.2.2 エージェントモデル

状態集合 $\mathcal{S} = \{s_0, s_1, \ldots, s_{24}\}$ とし，図 1(b) に示す通り，Stag か Rabbit への相対距離を用いて定義する．**行動は**，行動集合 $\mathcal{A} = \{a_{\rm s}, a_{\rm r}, a_{\rm w}\}$ のいずれかを選択する．ここで，$a_{\rm s}$ は Stag に，$a_{\rm r}$ は Rabbit に，それぞれ 1 マス近づき，$a_{\rm w}$: その場にとどまる行動である．**報酬**は，表 1(b) の利得行列で定義された値を報酬 r として与える．Rabbit，または， Stag の番地に次のステップも留まっていた場合, 獲得とみなして報酬が与えられる．次ステップで同番地から離れた場合は捕獲とはみなさない．Rabbit は単独で捕獲できるが，Stag は 2 エージェントが同時に捕獲行動を選択する必要がある．

2 つのエージェントが状態入力から行動を出力するまでを 1 ステップとし，両エージェントが Stag か Rabbit を捕獲するまでを 1 エピソードとする．ただし，一方が Rabbit を捕獲し，他方が Stag が追い続けて捕獲に至らない場合は 100,000 ステップで打ち切る．新しいエピソードは，再び初期配置から開始し，1 試行はエピソードを 30000 回繰り返しとする．

2.2.3 予備実験：Q 学習の報酬設定

2 つのゲームついて，各エージェントが Q 学習で方策を得る場合を図 2(a),(b) を用いて考察する．Q 学習における学習率 $\alpha = 0.1$, 割引率 $\gamma = 0.9$，行動選択には $\epsilon = 0.1$ の ϵ-greedy 法を用いた．

Coordination game:図 2(a) は，エージェント数 $N = 100$, 10×10 の二次元格子環境で，近傍エージェントとの距離 $D = 1, 10$ の平均獲得利得である．乱数種の異なる 100 試行実験し，その平均値である．近傍エージェント数 n_D は $n_1 = 4$, $n_{10} = 99$ であるが，局所性が強い状況下での学習では全エージェントが同一の均衡に収束することが困難であることが確かめられた．

Stag-Hunt game:図 2(b) は，Rabbit 捕獲によって得られる報酬を 10.0 としたとき，Stag 捕獲で与える報酬値 v_s(横軸) とその期待報酬値 (縦軸) を示している．この結果が示す通り，表 1(b) の利得値を報酬として与えても Stag 獲得には至らないことが確かめられた．したがって，Stag に付与する報酬値は，Stag を獲得しうる確率に基づいた期待報酬値を反映した設計が必要となる．

3. 逆強化学習による報酬関数の推定

逆強化学習 (Inverse Reinforcement Learning; IRL) を用いて報酬関数を推定し，これを用いてエージェントを望ましい均衡へと誘導させる方法を提案する．本章では逆強化学習とその導入方法について説明する．

3.1 関連研究

強化学習は環境から与えられる報酬に基づいてエージェントが自律的に行動ルールを獲得する手法であり，制御プログラムの自動化・省略化，ハンドコーディングよりも優れた解の発見が期待できるという利点がある．一方，学習性能が報酬の与え方に大きく依存する，専門家の技をロボットに実装するための効果的な報酬の与え方が分からないといった問題があり，報酬設計法の確立が課題として挙げられる [Russell 98].

逆強化学習は，Russell[Russell 98] によって最適な行動系列や環境モデルを所与として報酬関数を求める問題として定義さ

れ，様々な手法が提案されている [Ng 00][Ng 04][Sriraam 10]. Ng ら [Ng 00] は有限状態空間を持つ環境に対しては線形計画法，無限の状態空間を持つ環境に対してはモンテカルロ法を用いて報酬関数を推定する手法を示し，Abbeel ら [Ng 04] は報酬関数を推定する過程で最適な方策を獲得する “Apprenticeship learning” (見習い学習) の手法を示した．

また，Natarajan ら [Sriraam 10] はマルチエージェント環境において複数の報酬関数を推定して系全体の挙動を制御する手法を提案した．本稿は系全体での制御法ではなく，個々のエージェントがそれぞれ報酬関数を推定し，個々の自律制御を目指す．したがって，Natarajan らの手法ではなく，Ng ら [Ng 00] の提案した有限状態空間の逆強化学習法 (3.2 節) を用いる．

3.2 有限状態空間の逆強化学習法

ある状態 s_m $(m = 1, 2, \ldots, M)$ における最適な行動を a_1 とし，式 (2) の線形計画問題を解くことによって報酬関数 R' を推定する．式 (2) において，報酬関数ベクトル $\boldsymbol{R'}$ は状態 s_m の報酬 r'_{s_m} で与えられ，式 (3) で表す．状態遷移行列 $\boldsymbol{P}_a$ は行動 a の状態遷移確率 $P^a_{ss'}$ で与えられる $M \times M$ 行列であり，状態 s_m から行動 a をとり $s_{m'}$ に遷移する確率を $P^a_{mm'}$ とすると，$\boldsymbol{P}_a$ は式 (4) で表される．また，$\boldsymbol{P}_a(m)$ は，$\boldsymbol{P}_a$ の第 m 行ベクトルを表す．λ はペナルティ係数であり，λ を大きくすると高い価値を持つ状態を抽出できる．$R'_{\max}$ (> 0) は報酬の制約として設定する値である．

$$\text{maximize}: \sum_{m=1}^{M} \min_{a \in \mathcal{A} \backslash a_1} \{(\boldsymbol{P}_{a_1}(m) - \boldsymbol{P}_a(m)) (\boldsymbol{I} - \gamma \boldsymbol{P}_{a_1})^{-1} \boldsymbol{R'}\} - \lambda ||\boldsymbol{R'}||_1 \quad (2)$$

$$\text{s. t.}: (\boldsymbol{P}_{a_1} - \boldsymbol{P}_a)(\boldsymbol{I} - \gamma \boldsymbol{P}_{a_1})^{-1} \boldsymbol{R'} \succeq 0 \quad \forall a \in \mathcal{A} \backslash a_1$$
$$|r_{s_m}| \leq R'_{\max} \quad m = 1, \ldots, M$$

$$\boldsymbol{R'} = (r_{s_1}, \ldots, r_{s_m}, \ldots, r_{s_M})^{\mathrm{T}} \ (M \times 1 \text{ ベクトル}) \quad (3)$$

$$\boldsymbol{P}_a = \begin{pmatrix} P^a_{11} & P^a_{12} & \cdots & P^a_{1m} & \cdots & P^a_{1M} \\ P^a_{21} & P^a_{22} & \cdots & P^a_{2m} & \cdots & P^a_{2M} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ P^a_{m1} & P^a_{m2} & \cdots & P^a_{mm} & \cdots & P^a_{mM} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ P^a_{M1} & P^a_{M2} & \cdots & P^a_{Mm} & \cdots & P^a_{MM} \end{pmatrix} \quad (4)$$

逆強化学習法では，状態遷移確率は所与である．しかし，マルチエージェント環境の多くは状態遷移確率 $P^a_{ss'}$ が未知であるため，逆強化学習法を適用するために，状態遷移確率を観測によって推定する必要がある．状態遷移確率を推定する計算方法については，(1) ベイズ推定法に基づく方法と (2) 観測データの割合によって求める方法がある．本稿では (2) を用いる．具体的には状態遷移確率 $P^a_{ss'}$ の推定値 $\hat{P}^a_{ss'}$ を式 (5) で求める．

$$\hat{P}^a_{ss'} = \begin{cases} \frac{C_{s'}}{C_a} \ (C_a \neq 0) \\ 0 \ (C_a = 0) \end{cases} \quad (5)$$

式 (5) より，この方法では観測されない状態遷移確率の推定値 $\hat{P}^a_{ss'}$ は全て 0 になり，逆強化学習による報酬関数の推定に影響を与える．

4. 提案モデル

4.1 エージェントの定義

逆強化学習を導入するために Master，Mediator，Citizen の 3 種類のエージェントを導入する．**Master**$(ma \in \mathcal{M}_a)$ は，最適な行動 a_1 をとるエージェント，**Mediator**$(me \in \mathcal{M}_e)$ は，Master の行動系列を観測し，逆強化学習によって推定した R' から得られる状態価値 $V(s)$ に基づいて行動するエージェント，**Citizen**$(c \in \mathcal{C})$ は，協調ゲームでは表 1(a)，スタグハントゲームでは表 1(b) で定義される報酬 r に基づいて Q 学習を行うエージェントである．

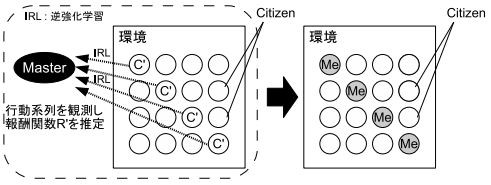


図 3: Induce convergence to the unique equilibrium via inverse reinforcement learning

4.2 逆強化学習の導入

問題 1: 協力ゲーム: 逆強化学習導入の流れを図 3 に示す．逆強化学習による統一行動への誘導は以下の 2 段階で行われる．

1. 環境から割合 p の $c' \in \mathcal{C}'$ $(\subseteq \mathcal{C})$ を選び ma と繰り返し協調ゲームを行わせる．c' は ma の行動系列から報酬関数 R' を推定する．
2. c' は R' から得られる状態価値 $V(s)$ に基づいて行動する me となり，c の学習に影響を与える．

問題 2: スタグハントゲーム: スタグハントゲームでは，Stag 捕獲を選択する Master と，Master の行動を観測する Citizen を考える．エージェント i を Master，エージェント j を Citizen として，Rabbit，Stag のどちらかを両エージェントが獲得するまでを 1 エピソードとする．

4.3 報酬関数 R' の推定

本提案手法は，逆強化学習を用いて，最適な状態に至るまでの行動系列を知っている Master の行動から，各状態の報酬関数を推定し，その後，この報酬関数に基づいてエージェントに強化学習させることによって，望ましい均衡解に収束させる方法である．

5. 報酬関数推定による均衡収束効果

5.1 問題 1:協調ゲーム (Coordination Game)

報酬関数：環境から割合 p で選ばれた c' と ma が協調ゲームを 10,000 ステップ繰り返し，c' が ma の報酬関数 R' を推定する．割引率 $\gamma = 0.9$，$R'_{\max} = 1.0$ とする．ペナルティ係数 $\lambda = 0, 4$ を用いる．

状態表現は表 2 の $l = 2$ とし，$t-1, t$ の自分と相手の行動を状態入力とする．ここではエージェント i が ma，j が c' とし，行動規則は，c' はランダムに行動するが，ma は 1 ステップ前の対戦相手の行動を選択，すなわち，統一行動を促すために c' の行動に合わせる行動を採る．また，全状態間の遷移を観測する必要があるため，初期状態は毎ステップランダムに生成する．報酬は与えない．推定した報酬関数 R' を図 4(a) に示す．横軸は状態 s_m，縦軸は推定報酬 r'_{s_m} である．$l = 2$，$\lambda = 0$ 場合どの状態からも遷移しない状態 $\{s_3, \ldots, s_6, s_{11}, \ldots, s_{14}\}$ の r'_{s_m} が 0，遷移が観測される状態 $\{s_1, s_2, s_7, \ldots, s_{10}, s_{15}, s_{16}\}$ の r'_{s_m} が 1 となった．さらに図 4(b) の $\gamma = 0.9$，$\lambda = 4$ では，時刻 $t-2$ で表 1 より 1 の報酬が得られたときに時刻 $t-1$ においても同じ行動をとる状態 $\{s_1, s_2, s_{15}, s_{16}\}$ が価値の高い状態として検出された．

図 4(b) は，$l = 2$ で観測される状態遷移例で，状態 s_t から矢印で示される次状態 s_{t+1} に遷移する様子を示している．遷移が観測されるグレーで表した状態 $\{s_1, s_2, s_7, \ldots, s_{10}, s_{15}, s_{16}\}$ が $\lambda = 0$ において $r'_{s_m} = 1$ として検出された．

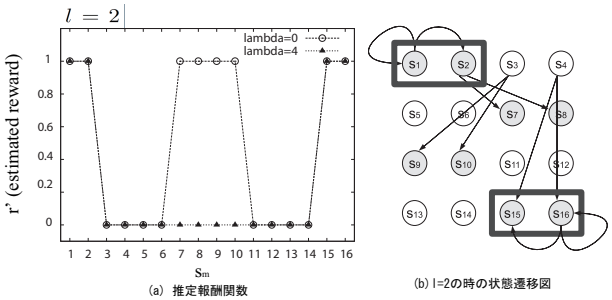


図 4: R': Estimated reward function of master's ($l = 2$)

収束効果：5.1 節で推定した報酬関数 R' を持つ $\mathcal{C}'$ を Mediator とし環境に導入した場合の，統一行動の誘導効果を考察する．実験対象問題は，2.2.3 節，図 2(a) での予備実験で，統一行動への収束に失敗した設定，エージェント数 $N = 100$，$D = 1$ の環境を用いる．Mediator の配置は図 5 右に示す 4 種類とし，Mediator の割合 p には $p = 0.00, 0.04, 0.25, 0.50, 1.00$ を用いる．

Master から推定した報酬関数 R' を持つ Mediator(me) は状態 s_m の価値関数の値 $V(s_m)$ を$V(s_m) = r'_{s_m}$ とし，$V(s_m)$ にしたがって greedy に行動する．Citizen(c) は Q 学習しつつ，ϵ-greedy 法（$\epsilon = 0.1$）で行動を選択する．状態表現は me, c ともに表 2，$l = 2$，ペナルティ係数 $\lambda = 4$ の結果を図 5 左に示す．乱数種の異なる 100 試行実験し，1 試行ごとに統一行動に収束したステップ数 T_e である．ここで，横軸は Mediator の割合 p, 縦軸は T_e を表す．$\lambda = 4$ では $1,000,000$ ステップまでに全ての試行で統一行動へと収束した．

5.2 問題 2:スタグハントゲーム (Stag-Hunt Game)

状態集合は図 1(b)，行動規則は， Master は，S 番地上以外なら a_{s}，S 番地上のとき a_{w} ，一方，Citizen は，2.2.2 節で示した行動集合に従う．協調ゲームと同様に全状態間の遷移を観測する必要があるため，初期状態は毎ステップランダムに生成する．報酬は与えない．

推定した報酬関数は，式 (2) のペナルティ関数 λ の値の変更によって目的関数が修正されるため, 制約条件式を満足させる解が異なる．λ の値を増加させると, 状態遷移確率の値が大きい上位の状態に対して推定報酬値が得られる．表 3 にいくつかの λ の下で得られた報酬値を用いた TD 学習によって Stag 獲得に成功した割合をまとめる．

例えば，λ=39.0 の時，報酬値が最も高い状態 S_{19} は「自分は Stag の番地の上, 相手は Rabbit の番地の上」である．. これは, Citizen がランダムに行動しているため, Master が S 番地上にいるにもかかわらず, Rabbit を獲得するときが状態 S_{19} であり, この状態への遷移回数が最も多いことから状態遷移確率が大きくなったためである．しかしこの値を用いて TD 学習すると，全試行で Rabbit 獲得を学習し, この報酬関数では Stag 獲得には至らない. つまり，状態 S_{19} の価値だけでは均衡収束に誘導するには不十分である．

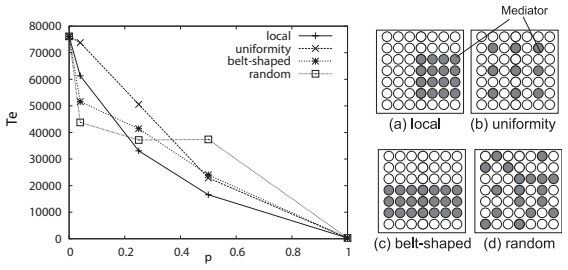


図 5: Average of required steps to be converged

表 3: Effect of introducing IRL

λ	報酬値が付与された状態 ID 図 1(b) の ID 参照	Stag 獲得 成功率 %
39.0	S_{19}	0.0
34.0	S_{19},S_{18}	100.0
23.0	S_{19},S_{18},S_{15}	100.0
10.0	S_{19},S_{18},S_{15} S_{17}	100.0
7.05	S_{19},S_{18},S_{15} S_{17},S_{16}	100.0
2.05	S_{19},S_{18},S_{15} S_{17},S_{16} S_4,S_3,S_2 S_1,S_0	99.0
1.05	S_{19},S_{18},S_{15} S_{17},S_{16} S_4,S_3,S_2 S_1,S_0,S_7,S_{12}	97.0
0.00	S_{19} ～S_0	94.0

6. まとめ

複数均衡が存在するマルチエージェント環境において，報酬に基づいて学習するエージェントに対して，同一均衡点に収束する行動を獲得させるための報酬関数推定法として逆強化学習によるモデリングを示した．本手法は線形計画法ベースにしており，ペナルティ係数の影響を受けるため，今後は非線形法を導入した拡張を課題とする．

参考文献

[Watkins 92] C. J. C. H. Watkins, P. D. Dayan：Q-learning, Machine Learning, Vol.8, 279/292(1992)

[Sen 07] S. Sen, S. Airiau：Emergence of norms through social learning, Proc. 20th International Joint Conference on Artificial Intelligence, 507/1512(2007)

[Mukherjee 08] P. Mukherjee, S. Sen, S. Airiau：Norm Emergence Under Constrained Interactions in Diverse Societies, Proc. 7th international joint conference on Autonomous agents and Multiagent systems, Vol.2, 779/786(2008)

[Russell 98] S. Russell：Learning agents for uncertain environments(extended abstract), Proc. 16th ICML, 278/287(1998)

[Ng 00] A. Ng and S. Russell：Algorithms for inverse reinforcement learning, in ICML(2000)

[Ng 04] P. Abbeel and A. Ng：Apprenticeship learning via inverse reinforcement learning, in ICML(2004)

[Sriraam 10] S. Natarajan, G. Kunapuli, K. Judah, P. Tadepalli, K. Kersting, and J. Shavlik：Multi-agent inverse reinforcement learning, in ICMLA 2010, 395/400, IEEE(2010)