軌跡に基づくエージェント固有の行動則とインセンティブの推定

2021年2月

千葉大学大学院融合理工学府

地球環境科学専攻都市環境システムコース

浪越　圭一

（千葉大学審査学位論文）

軌跡に基づくエージェント固有の行動則とインセンティブの推定

2021年2月

千葉大学大学院融合理工学府

地球環境科学専攻都市環境システムコース

浪越　圭一

論文要旨

近年，自律ロボットや自動運転車といった人工物の社会実装が進む中で，望ましい流れを実現するために，人の振舞いを予測し，必要に応じて人や他の人工物を制御する方法が求められている．しかし，人の振舞いを再現する行動則や自律主体を制御するインセンティブを個々に設計するのは，その組み合わせが膨大となり人手による設計は困難である．そこで本論文は，現実の振舞いや望ましい振舞いを軌跡として与え，その軌跡からエージェント固有の行動則やインセンティブを自動推定する枠組みをまとめる．論文は大きく二つ研究からなる．まず，軌跡に基づく行動則推定法として，行動則が同一なエージェントのグループとその行動則を進化計算により推定する方法を提案する．そして計算機実験により，エージェント 20 体の群衆シミュレーションにおいて，個体差を考慮しない従来法と比べ，個体差の有無にかかわらず軌跡を再現する行動則が得られることを示す．次に，軌跡に基づくインセンティブ推定法として，マルチエージェント逆強化学習によるアプローチを提案する．まず，既存法である座標降下法に基づく解法について，その妥当性と並列化による学習速度の改善可能性を実験的に示す．次に，報酬と行動則が同一なエージェントのグループとその報酬を同時推定するため，EM アルゴリズムと座標降下法の解法を組み合わせた推定法を提案し，エージェント 3 体の環境でその妥当性を示す．

目 次

第1章　序論

1.1　研究背景

歩行者や群衆といった人流や，自動車からなる交通流など，人の流れは複数の自律主体（エージェント）からなる Multi-Agent System（MAS）であり，その流れの最適化には予測と制御が重要な研究課題である．何故なら，各エージェントがもつ局所的な目的は，系全体を最適化したい設計者の目的と必ずしも一致しないからである．例えば，MAS のベンチマークである狭路すれ違い問題がある．この問題では，2 体のエージェントが，狭路の両端から，それぞれ反対側にある目標へ移動する．両方のエージェントに目標を達成させるには，いずれかのエージェントが狭路の脇によって道を譲る必要ががる．しかし，各エージェントが移動時間を最小化すると，譲り合いが生じない．このように，各エージェントのもつ目的によっては，両方のエージェントに目標を達成させたい設計者の目的と一致しない場合がある．そのため，流れの最適化には，事前に各エージェントの行動を予測し，望ましくない流れが予想される場合には，エージェントに適切なインセンティブを与えることで制御する必要がある．

MAS における行動予測やその制御に関する研究は多く存在し，前者は Agent-Based Model（ABM），後者はマルチエージェント強化学習（MARL）が知られる．ABM は，1990 年代に発達したシミュレーション法であり [Heath 09]，近年では経済学，都市・建築学，交通工学，認知科学など，幅広い領域への適用が報告されている [Macal 18]．ABM の特徴は，系全体の振舞いをトップダウンに記述するのではなく，個々の行動則，すなわちエージェントの観測に対する行動のマッピング関数から，系全体の振舞いを導く点にある．そのため，系の振舞いを直接記述するのが難しい対象を扱える．一方，MARL は強化学習 [Sutton 18] の発展を契機として 1990 年代から始まったパラダイムであり [荒井 98]，近年では，深層学習の発達に伴い 2 人ゲームである囲碁やポーカー，チーム対戦型の DOTA2 や StarCraftII といった勝敗の明確なゲームへの適用が報告されている [Hernandez-Leal 19]．MARL は，エージェントの振舞いに対するインセンティブを報酬関数として記述することで，各エージェントはその報酬を最大化する行動則を学習する．

ABM や MARL における課題の一つは，現実の振舞いや望ましい振舞いの獲得に，行動則や報酬関数の試行錯誤的設計を必要とする点にある．設計が困難な理由の一つは，個々のエージェントの意思決定が他エージェントの行動に左右される点にある．例えば，エージェントの目的が他エージェントと競合する場合は，相手行動による目的の阻害を防ぐために，

目的が他エージェントと同一の場合は，効率的な負荷分散を実現するために，いずれも相手行動を予測する必要がある．この相互作用に存在により，ABM や MARL で獲得したい振舞いが自明であっても，個々のエージェントの行動則や与えるべき報酬関数を直接導くことができない．

人手による試行錯誤的な設計を回避するため，近年は，実際の人の振舞いから行動則を推定するデータ駆動型 ABM や模倣学習，望ましい振舞いから報酬を推定するマルチエージェント逆強化学習が発展しつつある．いずれの手法も，エージェントの軌跡から行動則や報酬を自動的に推定するため，人手による試行錯誤を軽減することが期待できる．しかし，既存の推定法では，エージェント間の個体差に関する議論は少ない．詳しくは 4 章で述べるが，いずれの既存法も全エージェントに共通，もしくは個々に異なる行動則や報酬を推定する．そのため，共通する行動則では再現できない課題や，個々の報酬では過学習を引き起こすといった課題が生じうる．よって，エージェントの軌跡から行動則や報酬を推定する手法に対し，エージェント間の個体差に関して議論の余地があるといえる．

1.2 研究目的

本論文の目的は，(1) ABM において現実の振舞いを再現するエージェントの行動則，(2) MARL において望ましい振舞いを得る報酬関数を，それぞれの振舞いを反映したエージェントの軌跡から自動推定することであり，そのうち特に，エージェント間に個体差のある環境を対象とする．

エージェント間の個体差は，一般的に，異質性（Heterogeneity）と呼ばれ，鳥や魚の群れなど，同質のエージェントからなる系は同種（Homogeneous），それ以外の場合は異種（Heterogeneous）として区別される [Dorri 18]．本論文においては，エージェントの行動則や報酬関数が異種の環境，特に，全エージェントで異なるのではなく，同種なエージェントと異種なエージェントが混在する環境を想定する．この想定は，年齢や性別といった属性，エージェント間の物理的距離によるグループごとに，同一の行動則や報酬関数をもつ可能性を反映したものである．実際に，社会心理学における火災時の人の反応行動として，所属するグループ全員の集合をまつ，社会的役割によって反応が異なる，といった説明がなされる [Yang 05] ことからも妥当な仮定といえる．そこで，同種・異種が混在する MAS を対象に，行動則や報酬関数の獲得を目指す．

本論文は，大きく二つの研究で構成される．

研究 1：群衆 Agent-based model における異種の戦略推定

一つめの研究では，ABM において望ましい振舞いを再現する行動則を得るため，群衆シミュレーションを対象に，エージェントの行動則にあたる戦略を軌跡から推定する．軌跡は，カメラや IoT（Internet of Things）デバイスにより，現実の人々の振舞いを観測した場合を

想定している．また，戦略は「エージェントがどの状態を目標とするのか」に関する決定基準を指す．

戦略を推定する既存研究には Zhong ら [Zhong 14] がある．この研究は，戦略を目標状態に対する評価関数として定義し，この関数を進化計算を用いて推定する方法を提案している．しかし,「全エージェントは同種の戦略を持つ」という前提があるため，推定される戦略（評価関数）は全エージェントに共通であり，異種の戦略推定は対象としていない．

そこで,「エージェントによって戦略が異なる」場合の推定を目的として，進化計算法のうち，複数の戦略が扱える枠組みである Automatically Defined Groups[Hara 99, 原 00] を導入することで，戦略の同じエージェントのグループと，各グループの従う戦略を推定する．

研究 2：マルチエージェント逆強化学習における異種の報酬推定

二つめの研究では，強化学習における報酬設計問題のアプローチである逆強化学習を，マルチエージェント系に導入する．提案法は，マルコフ決定過程を前提とした逆強化学習を，マルコフゲームに拡張したマルチエージェント逆強化学習（MAIRL：Multi-agent Inverse Reinforcement Learning）として位置付けられる．本論文では，異種の報酬推定の課題の一つである学習速度の改善と，報酬が同一なグループ構造を報酬と同時に推定する方法の二つを提案する．

研究 2-1：並列座標降下法による報酬の学習速度改善

軌跡からその軌跡を生成する報酬関数を推定するマルチエージェント逆強化学習は，推定報酬に対する最適方策を計算するマルチエージェント強化学習（MARL）問題を内包する．MARL は，同時学習問題により収束が難しい課題と，次元の呪いによりエージェント数に対して計算量が増大する課題がある．

同時学習問題を回避する従来法には，座標降下法を用いた MAIRL[Ziebart 10b, Yang 20b] がある．この手法では，最適方策の計算をエージェント 1 体に限定し，他エージェントの方策を固定することで単一エージェントの逆強化学習問題に帰着している．一方で，次元の呪いは生じるため，各更新をエージェントごとに巡回更新するごとに，探索空間の大きな強化学習を何度も解く必要がある．

そこで，エージェント数に対する学習速度の悪化を軽減する方法として，並列座標降下により更新を並列化することが可能か示す．

研究 2-2：グループ構造と報酬の同時推定

研究 2-1 を含む従来の MAIRL は，zero-sum 報酬 [Lin 18, Wang 18]，全エージェントに共通の報酬 [Šošić 17]，個々に異なる報酬 [Yu 19, Wei 19] のいずれかを推定時に仮定する．しかし，軌跡中のエージェントが共通の報酬を最大化しているのか，個々に異なる報酬を最

大化しているのか，といった報酬の構造は，軌跡間で必ずしも同一でないうえに，軌跡から直接仮定するのは難しい．一方で，各軌跡について個々の報酬を求めるのは冗長な表現となる．

そこで，エージェントの行動則と報酬が同一なエージェントのグループを導入し，軌跡がグループ構造ごとの生成分布の混合分布で生成された，という仮定から，EM アルゴリズムによりグループ構造と報酬を同時推定する方法を提案する．

1.3 論文構成

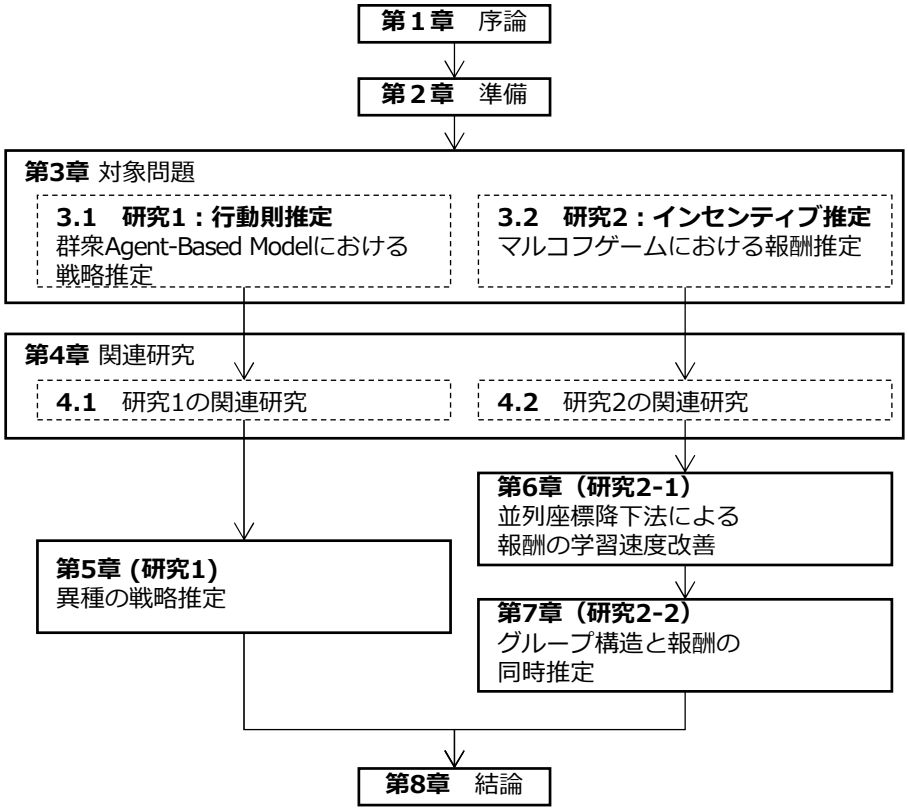


図 1.1: 論文の構造

図 1.1 に本論文の構造を図示する．まず，準備として 2 章で基礎事項を述べる．つぎに，3 章に対象問題を示し，4 章で関連研究を示す．そして，ABM におけるエージェントの行動則推定を 5 章，MAIRL において，並列座標降下法を用いた報酬の学習速度改善を 6 章，EM アルゴリズムを用いたグループ構造と報酬の同時推定を 7 章でそれぞれ述べ，最後に，8 章で結論を述べる．

第2章　準備

本章では，本論文を読み進めるにあたり必要となる基礎事項として，5 章で用いる進化計算の遺伝的アルゴリズム，遺伝的プログラミング，群衆モデルの代表的なモデルである Social force model[Helbing 00]，6 章，7 章で用いるマルコフゲーム，逆強化学習についてそれぞれ述べる．

2.1　遺伝的アルゴリズム

遺伝的アルゴリズム（GA：Genetic Algorithm）[BoussaïD 13, Eiben 15] は，遺伝子によって生物が環境に適応しながら進化する過程を模倣した最適化法である．GA により獲得する解は，必ずしも大域的最適解ではないものの，ランダムサーチの苦手とする解が疎な場合や，山登り方の苦手とする多峰性のある場合に，比較的性能の高い解を短時間で見つけることが期待できる [哲也 93]．

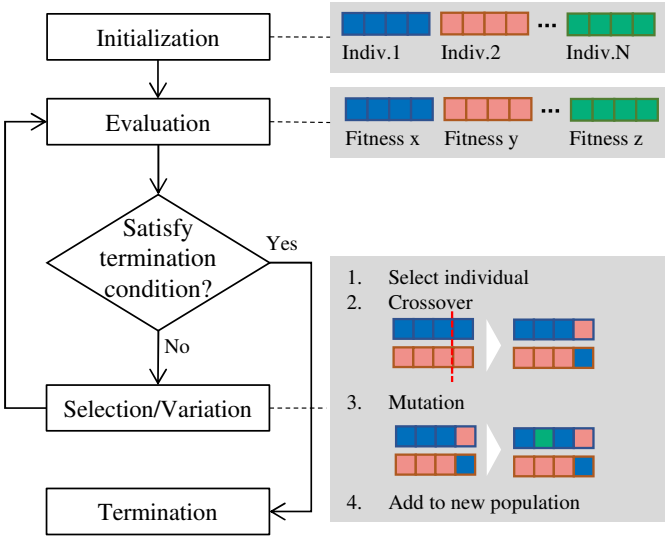


図 2.1: 遺伝的アルゴリズムの更新手順

図 2.1 に，GA の探索手順を示す．はじめに，ランダム生成された解候補（個体）からなる集団を第 1 世代とする．つぎに，集団の各個体を適応度関数によって評価し，打ち切り条件を満たさない場合は，遺伝的オペレータによって第 2 世代の集団を生成する．以降は，適応度関数による評価と遺伝的オペレータの実行を繰り返し，収束または打ち切り条件を満たすとき探索を打ち切る．

GA を用いる場合，対象問題に合わせて次の (1) から (5) を定める必要がある．

(1) 遺伝子表現

遺伝子表現とは,「対象問題における解の表現」を「GA における解の表現」へ置き換えたもののことを指す．前者は表現型，後者は遺伝子型と呼ばれる．GA では遺伝子型に 1 次元配列が用いられ，配列の各要素を遺伝子，配列上の遺伝子の位置を遺伝子座と呼ぶ．

(2) 適応度関数

適応度関数とは，対象問題における個体の適応度合いを評価する関数であり，最適化問題においては目的関数を指す．

(3) 遺伝的オペレータ

遺伝的オペレータとは，複数の個体からなる集団について，既に生成された集団から新しい世代の集団を生成する操作を指す，遺伝的オペレータには，選択，交叉，突然変異などが用いられる．交叉および突然変異は，交叉確率および突然変異確率に基づいて実行される．遺伝的オペレータにはさまざなな手法が存在するため，以下では論文中で用いる手法のみ示す．

選択：集団から，次世代の集団に加える個体を選択する．**トーナメント選択**は，集団からランダムに n 個の個体を選び，その中で最も適応度が最も高い個体を選択する．n はトーナメントサイズと呼ばれるパラメータである．また，$n = 2$ のトーナメント選択は，バイナリトーナメント選択と呼ばれる．**エリート選択**は，集団の中で適応度が最も高い個体を，他の遺伝的オペレータを作用させず，次世代にそのまま残す

交叉：選択された二つの個体に対して，個体の遺伝子を組み替える．**一点交叉**は，両個体で共通するランダムな切れ目を一つ選び，切れ目の左右の遺伝子を個体間で入れ替える．**二点交叉**は，両個体で共通するランダムな切れ目を二つ選び，切れ目同士の間に位置する遺伝子を個体間で入れ替える．

突然変異：個体の一部をランダムに生成した遺伝子で置き換える．

(4) パラメータ

パラメータには，集団を構成する個体数，遺伝的オペレータのパラメータがある．

(5) 打ち切り条件

適応度が最も高い個体が予め決められた閾値を満たす場合か，世代数が上限に達した場合に探索を打ち切る．

2.2 遺伝的プログラミング

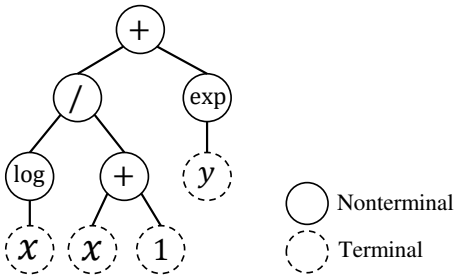


図 2.2: 遺伝的プログラミングにおける遺伝子型の例（関数 $\frac{\log x}{x+1} + \exp y$ の遺伝子型を表す.）

遺伝的プログラミング (GP：Genetic Programming）とは，木構造やネットワーク構造の遺伝子型を扱う進化計算の総称で，関数や概念，関係を扱うことの出来る手法として，1次元配列を扱う GA と区別される．本論文では，関数表現のみ扱うことから，以下では遺伝子型が木構造の場合についてのみ述べる．

GP は，GA と次の点が異なる．

(1) 遺伝子表現

図 2.2 に遺伝子型の例を示す．木構造で関数を表現する場合，演算子や変数を各ノードにラベル付した順序木で表す．順序木とは，各ノードの中で，他と区別された一つの頂点（根）を持ち，任意のノードのもつ子の順序に意味があるデータ構造を指す．終端（葉）ノードには変数や定数，非終端ノードには関数や演算子がラベル付けされ，前者を非終端記号集合，後者を終端記号集合と呼ぶ．そのため，遺伝子型の設計とは，非終端記号集合と終端記号集合を定めることを意味する．

(2) 遺伝的オペレータ

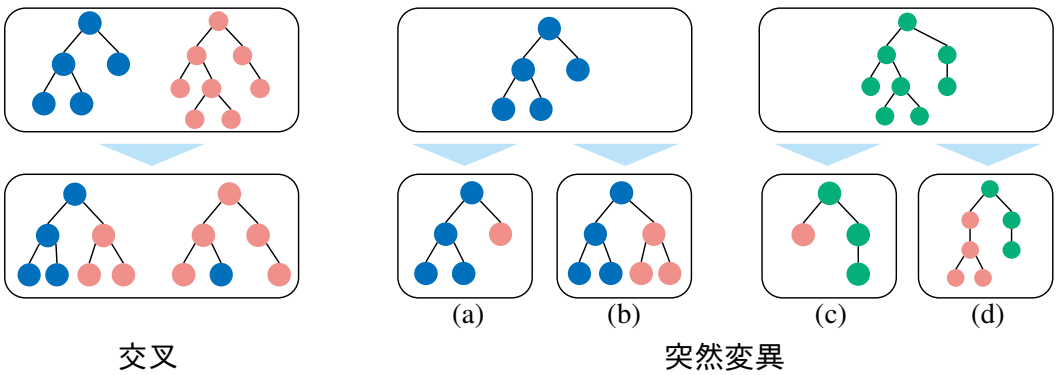


図 2.3: 遺伝的プログラミングにおけるオペレータ

交叉：図 2.3 に交叉の例を示す．選択された二つの個体に対し，個体の部分木同士を組み替える．組み替える部分木は，個体毎にランダムなノードを選択し，そのノードを根とする部分木を用いる．

突然変異：ランダムなノードを選び，そのノードを根とした部分木を，ランダムに生成した部分木に置き換える．ただし，図 2.3 に示すように，突然変異が起こるノードと置き換わるノードの種類によって次のように分類される．[伊庭 96]

(a) 終端ノードから非終端ノードへの突然変異：新しい部分木の生成

(b) 終端ノードから終端ノードへの突然変異：ノードラベルの付け替え

(c) 非終端ノードから終端ノードへの突然変異：部分木の削除

(d) 非終端ノードから非終端ノードへの突然変異：新しい非終端ノードと古い非終端ノードの子の数が等しい場合にはノードラベルの付け替え，異なる場合は部分木の生成・削除

アルゴリズム

Algorithm 1 Genetic programming

Input: Population size N, Selection parameter p^{exp}, Crossover rate p^{cross}, Mutation rate p^{mutation}

$\mathcal{P}_1 \leftarrow$ randomly generated N individuals ▷ 初期化
for generation $i = 1, 2, \cdots$ **do**
  Calculate fitness value of $n \in \mathcal{P}_i$ ▷ 評価
  if the best individual of $\mathcal{P}_i$ satisfies the termination condition **then**
    Break
  end if
  Add the fittest individual to $\mathcal{P}_{i+1}$ ▷ エリート選択
  while $|\mathcal{P}_{i+1}| < N$ **do**
    if $x > p^{\text{new}}, x \sim \text{uniform}(0, 1)$ **then**
      $n_1, n_2 \leftarrow \text{selection}(\mathcal{P}_i)$ ▷ 選択
      $n_1, n_2 \leftarrow \text{crossover}(n_1, n_2, p^{\text{cross}})$ ▷ 交叉
      $n_1, n_2 \leftarrow \text{mutation}(n_1, n_2, p^{\text{mutation}})$ ▷ 突然変異
      Add n_1 and n_2 to $\mathcal{P}_{i+1}$
    else
      Add randomly generated individual to $\mathcal{P}_{i+1}$
    end if
  end while
end for

Algorithm 1 に文献 [Segaran 08] に基づく GP の疑似アルゴリズムを示す．ここで，$\mathcal{P}_i$ は世代 i の集団，N は個体数，$p^{\text{crossover}}$ は交叉確率，p^{mutation} は突然変異確率，p^{new} は次

世代にランダムな木を追加する確率をそれぞれ表す．選択関数 selection($\mathcal{P}_i$) は，世代 i の集団から次世代の個体の元を選び出す．本論文中では，2.1 節で述べたトーナメント選択やバイナリトーナメント選択が用いられる．交叉関数 crossover($n_1, n_2, p^{\text{cross}}$) は，まず，$n_1$ の木のノードを一つずつ参照しながら確率 $p^{\text{crossover}}$ でノードを選択，次に，n_2 のノードをランダムに選択，最後に，選ばれたノードを根とする部分木を交換する．突然変異関数 mutation($n_1, n_2, p^{\text{mutation}}$) は，$n_1$，$n_2$ のそれぞれで，木のノードを一つずつ参照しながら確率 p^{mutation} でノードを選択，選択されたノードを根とする部分木をランダムに生成した木に置き換える．

2.3 Social Force Model

Social Force Model（SFM）[Helbing 00] は，目的地からの引力と，障害物や他のエージェントからの斥力をそれぞれ仮想的に作用させ，歩行者の振舞を再現するモデルである．そのため，出口付近で発生する「つまり」といった振舞の再現に用いられる．

SFM において，エージェント i は式 (2.1) に定義する方程式に従って移動する．

$$\mathbf{f}_i = m_i \frac{\mathrm{d}\mathbf{v}_i}{\mathrm{d}t} = m_i \frac{v_i^0(t)\mathbf{e}_i^0(t) - \mathbf{v}_i(t)}{\tau_i} + \sum_{j(\neq i)} \mathbf{f}_{ij} + \sum_W \mathbf{f}_{iW} \tag{2.1}$$

ここで，m_i はエージェントの質量，$\frac{\mathrm{d}\mathbf{v}_i}{\mathrm{d}t}$ は加速度，右辺第 1 項の $v_i^0(t)$ は目標速度，$\mathbf{e}_i^0(t)$ は目標方向，$\mathbf{v}_i(t)$ は現在の速度，τ_i は加速にかかる時間，第 2 項の $\mathbf{f}_{ij}$ はエージェント $j(\neq i)$ から受ける反発力，第 3 項の $\mathbf{f}_{iW}$ は壁 W から受ける反発力である．

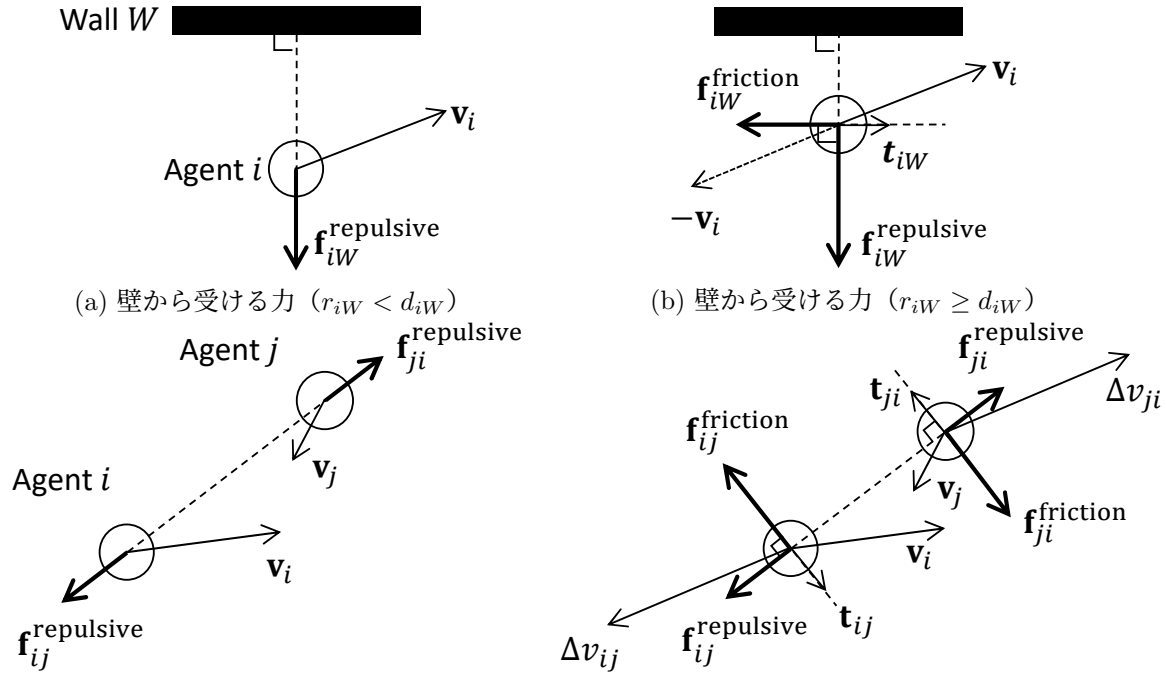
また，右辺第 2 項の $\mathbf{f}_{ij}$ と第 3 項の $\mathbf{f}_{iW}$ は式 (2.2)，式 (2.3) でそれぞれ表される．

$$\mathbf{f}_{ij} = \underbrace{\left\{ \alpha_i \exp\left(\frac{r_{ij} - d_{ij}}{\beta_i} \right) + \gamma g(r_{ij} - d_{ij}) \right\} \mathbf{n}_{ij}}_{\text{repulsive force } \mathbf{f}_{ij}^{\text{replusive}}} + \underbrace{\kappa g(r_{ij} - d_{ij}) \Delta v_{ij}^t \mathbf{t}_{ij}}_{\text{friction force } \mathbf{f}_{ij}^{\text{friction}}} \tag{2.2}$$

$$\mathbf{f}_{iW} = \underbrace{\left\{ \alpha_i \exp\left(\frac{r_i - d_{iW}}{\beta_i} \right) + \gamma g(r_i - d_{iW}) \right\} \mathbf{n}_{iW}}_{\text{replusive force } \mathbf{f}_{iW}^{\text{replusive}}} + \underbrace{\kappa g(r_i - d_{iW}) \Delta v_{iW}^t \mathbf{t}_{iW}}_{\text{friction force } \mathbf{f}_{iW}^{\text{friction}}} \tag{2.3}$$

$$g(x) = \begin{cases} 0 & (x < 0) \\ x & (otherwise) \end{cases} \tag{2.4}$$

ここで，$\mathbf{r}_i$ はエージェント i の座標，r_i はエージェント i の半径，$d_{ij} = ||\mathbf{r}_i - \mathbf{r}_j||$ はエージェント i, j の距離，$r_{ij} = r_i + r_j$ はエージェント i, j の半径の合計，$\mathbf{n}_{ij} = (n_{ij}^1, n_{ij}^2) = (\mathbf{r}_i - \mathbf{r}_j)/d_{ij}$ はエージェント j から i への単位ベクトル，$\mathbf{t}_{ij} = (-n_{ij}^2, n_{ij}^1)$ は $\mathbf{n}_{ij}$ を 90 度回転した単位ベクトル，$\Delta v_{ij}^t = (\mathbf{v}_j - \mathbf{v}_i) \cdot \mathbf{t}_{ij}$ はエージェント i, j 間の相対速度の接線ベクトル $\mathbf{t}_{ij}$ への正射影ベクトル，$\Delta v_{iW}^t = (0 - \mathbf{v}_i) \cdot \mathbf{t}_{iW}$ はエージェント i と壁 W 間の相対速度の接線ベクトル $\mathbf{t}_{iW}$ への正射影ベクトル，関数 $g(x)$ は式 (2.4) で表される．また，$\alpha_i, \beta_i, \gamma, \kappa$ は定数である．



(a) 壁から受ける力（$r_{iW} < d_{iW}$）

(b) 壁から受ける力（$r_{iW} \geq d_{iW}$）

(c) 他エージェントから受ける力（$r_{ij} < d_{ij}$）

(d) 他エージェントから受ける力（$r_{ij} \geq d_{ij}$）

図 2.4: Social Force Model においてエージェントに作用する力の方向（各力の方向をのみを示すため，エージェントの半径や各ベクトルの長さ，エージェントや壁の位置関係は正確ではない．特に，図 2.4(b) や図 2.4(b) に示す場合は，実際にはエージェントと壁・他エージェントは接触している．）

図 2.4 に反発力と摩擦力の概要図を示す．エージェントが他のエージェントか壁に接触していないとき，第 1 項のみが反発力として作用する．接触している時は，第 1 項に加えて第 2 項の非常に大きな反発力と，第 3 項の摩擦力が加えられる．摩擦力は，エージェントと他エージェント，もしくはエージェントと壁ごとの接線ベクトルに沿って作用し，作用方向は相対速度の接線ベクトルへの正射影方向となる．そして，式 (2.1) に従い，全てのエージェントと壁に対して総和を計算し，現在の速度ベクトルと目標速度ベクトルとの差を加えてエージェント i に加わる力 $\mathbf{f}_i$ とする．

Algorithm 2 に SFM を用いて時刻 t から $t + \Delta t$ にエージェントの座標を更新する疑似アルゴリズムを示す．Δt 秒後のエージェントの座標は式 (2.1) の微分法方程式を解くことで求まることから，オイラー法を用いて更新する．ただし，オイラー法における刻み幅 Δt を一定にした場合，エージェント間，またはエージェントと障害物間の距離が限りなく 0 に近づき，想定外の振舞を示すことがある．よって，文献 [SFModelSup] に基づき，Algorithm 2 の 4 から 7 行目で Δt を速度の増分が一定値以下になるように調整している．

Algorithm 2 Agent coordinate update by social force model

1: **for** $i \in [1, N]$ **do**
2: $\quad \mathbf{a}_i(t + \Delta t) \leftarrow \dfrac{v_i^0(t)\mathbf{e}_i^0(t) - \mathbf{v}_i(t)}{\tau_i} + \dfrac{\sum_{j(\neq i)} \mathbf{f}_{ij}}{m_i} + \dfrac{\sum_W \mathbf{f}_{iW}}{m_i}$ ▷ Update acceleration
3: **end for**
4: $\Delta t \leftarrow 0.01$
5: **while** $\max_i |\mathbf{a}_i(t + \Delta t)|\Delta t \geq 0.01$ **do** ▷ Adjust time increment
6: $\quad \Delta t \leftarrow 0.95\Delta t$
7: **end while**
8: **for** $i \in [1, N]$ **do**
9: $\quad \mathbf{v}_i(t + \Delta t) \leftarrow \mathbf{a}_i(t + \Delta t)\Delta t + \mathbf{v}_i(t)$ ▷ Update velocity
10: $\quad \mathbf{x}_i(t + \Delta t) \leftarrow \mathbf{v}_i(t + \Delta t)\Delta t + \mathbf{x}_i(t)$ ▷ Update coordinate
11: **end for**
12: $t \leftarrow t + \Delta t$

表 2.1 に SFM のパラメータを示す．各パラメータは文献 [Helbing 00] の値を用いている．

表 2.1: Social force model におけるパラメータと設定値

Parameter	Value
m_i	80[kg]
V_i^0	1[m/s]
τ_i	0.5[s]
α_i	2×10^3[N]
β_i	0.08[m]
γ	1.2×10^5[kg/s^2]
κ	2.4×10^5[kg/ms]
r_i	$[0.25, 0.35]$[m]

2.4 マルコフゲーム

エージェント N 体のマルコフゲームは $< \mathcal{S}, \mathcal{A}, P, \boldsymbol{r}, P_0, \gamma >$ の組で表される確率モデルである [Sigaud 10]．$\mathcal{S}$ は離散状態空間，$\mathcal{A} = \mathcal{A}_1 \times \cdots \times \mathcal{A}_N$ は全エージェントの離散行動空間，$P : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0, 1]$ は状態遷移確率，$r_i : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ はエージェント i の報酬，P_0 は初期状態分布，γ は割引率をそれぞれ表す．また，エージェント i の方策は $\pi_i : \mathcal{S} \times \mathcal{A}_i \to [0, 1]$ と表す．以下では，エージェント i を除いた他エージェントの集合を $-i$ と表し，全エージェントについて組み合わせた変数ベクトルは太文字で表す．

2.5 逆強化学習

逆強化学習（IRL : Inverse Reinforcement Learning）は，報酬を除いた（報酬が未知の）マルコフ決定過程 $< \mathcal{S}, \mathcal{A}, P, P_0, \gamma > \backslash r$ として表され，エージェントが生成した軌跡 $\tau = (s_t, a_t)_{t=0}^T$ の集合から報酬を推定する枠組みである．軌跡は割引累積報酬の期待値を最大化する（準）最適方策から生成されたとみなすことから，軌跡を生成したエージェントをエキスパートと呼ぶ．

IRL は，エキスパート軌跡を生成する方策と，その方策を得られる報酬が一意に定まらない不良設定問題であり，Russell による定式化 [Russell 98] 以降，不良設定に対するいくつかのの定式化 [Ng 00, Ziebart 08, Ramachandran 07] が提案されている [Zhifei 12, Arora 18]．これらの IRL のうち，本論文は，状態遷移確率が未知，かつ，報酬に関する事前知識を必要としない最大エントロピー原理に基づく Maximum Discounted Causal Entorpy IRL（MDCE IRL）[Zhou 18] を用いる．

MDCE IRL は，特徴ベクトル $\boldsymbol{f} : \mathcal{S} \times \mathcal{A} \to \mathbb{R}^k$ の割引期待値がエキスパート方策 π^E と一

致する方策 π の獲得問題として，式 (2.5) から式 (2.8) で定式化される．

$$\max_{\pi} \quad \alpha H(\pi) \tag{2.5}$$

$$\text{s.t.} \quad \overline{\boldsymbol{f}}_{\pi^E} = \overline{\boldsymbol{f}}_{\pi} \tag{2.6}$$

$$\pi(a|s) \geq 0 \quad \forall a \in \mathcal{A}, s \in \mathcal{S} \tag{2.7}$$

$$\sum_{a \in \mathcal{A}} \pi(a|s) = 1 \quad \forall s \in \mathcal{S} \tag{2.8}$$

ここで，式 (2.5) は式 (2.9) で定義される方策 π のエントロピー $H(\pi)$ に係数 α を乗じた値[1] である．また，式 (2.6) は式 (2.10) の割引特徴期待ベクトル $\overline{\boldsymbol{f}}_{\pi}$ を一致させる制約を，式 (2.7)，式 (2.8) は，それぞれ方策に関する制約を表す．$\overline{\boldsymbol{f}}_{\pi^E}$ はエキスパート軌跡集合の平均値で近似する．

$$H(\pi) \triangleq \mathbb{E}_{P_0,P,\pi}\left[\sum_{t=0}^{\infty} -\gamma^t \log \pi(a_t|s_t)\right] \tag{2.9}$$

$$\overline{\boldsymbol{f}}_{\pi} \triangleq \mathbb{E}_{P_0,P,\pi}\left[\sum_{t=0}^{\infty} \gamma^t \boldsymbol{f}(s_t, a_t)\right] \tag{2.10}$$

MDCE IRL のラグランジュ緩和問題は式 (2.11) で表される．

$$\min_{\boldsymbol{\theta}} L(\boldsymbol{\theta}) = \min_{\boldsymbol{\theta}} \left[\max_{\pi} \alpha H(\pi) + \boldsymbol{\theta}^\top \overline{\boldsymbol{f}}_{\pi}\right] - \boldsymbol{\theta}^\top \overline{\boldsymbol{f}}_{\pi^E} \tag{2.11}$$

MDCE IRL と緩和問題の間には強双対性が成り立ち，ラグランジュ係数 $\boldsymbol{\theta} \in \mathbb{R}^k$ と特徴ベクトルの内積は報酬 $r(s,a) = \boldsymbol{\theta}^\top \boldsymbol{f}(s,a)$ に一致する [Zhou 18]．式 (2.11) における報酬の重み $\boldsymbol{\theta}$ の最小化問題は勾配 $\nabla L(\boldsymbol{\theta}) = \overline{\boldsymbol{f}}_{\pi} - \overline{\boldsymbol{f}}_{\pi^E}$ による勾配降下法を用いて解く．一方，式 (2.11) における方策 π の最大化問題は Inner Loop と呼ばれ，割引累積報酬の期待値と方策のエントロピーを最大化する最大エントロピー強化学習 [Haarnoja 17] 問題と一致する．ここでの最適方策，行動価値関数，状態価値関数はそれぞれ式 (2.12) から式 (2.14) で表される．

$$\pi^*(a|s) = \exp\left(\frac{1}{\alpha}\left(Q^*(s,a) - V^*(s)\right)\right) \tag{2.12}$$

$$Q^*(s,a) = \boldsymbol{\theta}^\top \boldsymbol{f}(s,a) + \gamma \sum_{s' \in \mathcal{S}} P(s'|s,a) V^*(s') \tag{2.13}$$

$$V^*(s) = \alpha \log \sum_{a \in \mathcal{A}} \exp\left(\frac{1}{\alpha} Q^*(s,a)\right) \tag{2.14}$$

[1] 方策のエントロピーにまつわる係数 α は，後述の式 (2.11) において α を除することで，報酬 $\boldsymbol{\theta}^\top \boldsymbol{f}(s,a)$ のスケールを調整するハイパーパラメータとみなせる．$\alpha \to 0$ のとき，式 (2.13)，式 (2.14) で表される Soft Bellman 方程式が Bellman 方程式と一致することが知られており [Ziebart 10a]，最大エントロピー強化学習問題が強化学習問題の一般化であることを意味している．これは，式 (2.12) に示すように，方策が行動価値関数のボルツマン分布で表され，α が温度パラメータに相当することからも明らかである．そのため，最大エントロピー強化学習問題の学習速度を調整するハイパーパラメータとして活用される [Haarnoja 17].

第3章　対象問題

1章で述べたように，本論文では，エージェント間に個体差のある環境において，エージェントの軌跡を再現する行動則推定とマルチエージェント強化学習において軌跡を獲得できるインセンティブ推定の二つに取り組む．本章では，各推定問題の対象問題を示す．

3.1　行動則推定：軌跡を再現する群衆 Agent-Based Model の戦略推定

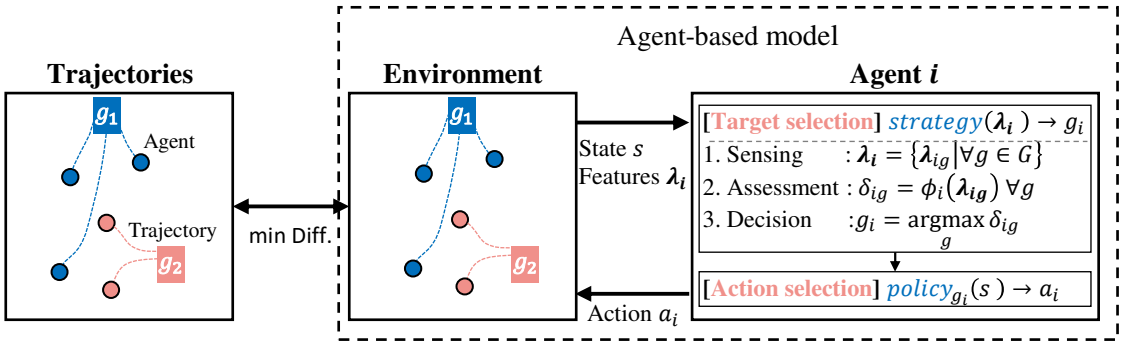


図 3.1: 群衆 Agent-Based Model における戦略推定

図 3.1 に対象問題を図示する．本論文では，行動則推定の具体例として，群衆を表す Agent-Based Model（ABM）において,「エージェントがどの状態を目標とするのか」に関する決定基準である戦略を推定する．

対象の群衆 ABM において，エージェントは，環境の観測に基づき，意思決定過程を経て行動を決定する．群衆 ABM における意思決定過程は,「戦略 (strategy) による目標状態の選択 (Target selection)」と,「方策 (policy) による目標状態に至る行動の選択 (Action selection)」の 2 段階から成ると仮定する．以下では，戦略と方策を説明する．

■戦略 (strategy)：エージェント i のもつ戦略は，観測した全目標状態 G の特徴量 $\boldsymbol{\lambda}_i$ から目標状態 $g_i \in G$ を出力する．目標状態 g_i の選択は以下の 3 段階からなる．

1. **観測 (Sensing)**: エージェント i の観測した目標状態 g を表す K 個の特徴量 $\boldsymbol{\lambda}_{ig} = \{\boldsymbol{\lambda}_{igk} \mid k \leq K, k \in \mathbb{N}\}$ を，全ての目標状態について観測する．

2. **評価 (Assessment)**: 各 g の評価値 δ_{ig} を式 (3.1) によって求める．ϕ_i は g に対する評

価関数を表す.

$$\delta_{ig} = \phi_i(\boldsymbol{\lambda}_{ig}) \tag{3.1}$$

3. **決定 (Decision)**: 評価値 δ_{ig} が最大となる目標状態 g を選ぶ.

■方策 (policy)：方策は，他エージェントや障害物といった環境の状態 s から衝突回避などを考慮した行動 a_i を出力する．行動 a_i は，戦略で選択した目標状態 g_i までの行動系列ではなく，g_i へ向かう 1 ステップ分の移動を指す.

前述の群衆 ABM に対し，本論文における行動則推定とは，全エージェントの軌跡，すなわち座標系列を所与として，軌跡に一致する戦略を推定する問題とする．ただし，戦略による選択は評価関数 ϕ_i によって決まることから，以下では ϕ_i を戦略と記す.

3.2 インセンティブ推定：軌跡を獲得するマルコフゲームの報酬推定

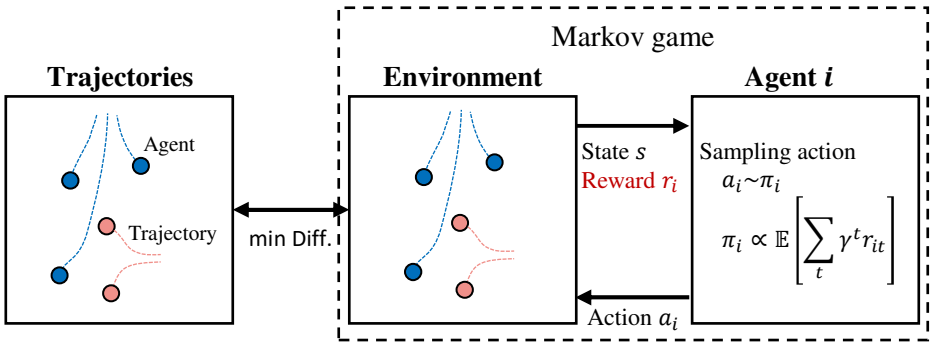


図 3.2: マルコフゲームにおける報酬推定

本論文におけるインセンティブ推定とは，図 3.2 に示す環境において，全エージェントの状態・行動の軌跡から，軌跡が一致する報酬関数を求めるマルチエージェント逆強化学習問題である．環境は 2.4 節に示した Markov game で記述される．各エージェントは，環境の状態と報酬を観測し，報酬の累積期待値を最大化するよう行動を決定する.

第4章　関連研究

本章では，軌跡に基づいた行動則やインセンティブの推定法を整理し，提案法の立ち位置を述べる．

4.1　行動則推定

図 4.1 に，マルチエージェント系を対象に行動則を推定する関連研究の位置づけを示す．関連研究は，エージェントの観測能力が完全な場合か部分的な場合の二つに大別できる．観測能力とは，環境の状態に対する観測可能性を表し，マルコフゲームの状態が観測できるとき[1]は完全，そうでない場合は部分的とする．観測能力が完全な場合の推定法は，敵対的学習を用いたマルチエージェント逆強化学習（MA-GAIL）[Song 18] とその発展法 [Jeon 20, Yang 20a]，または，4.2.2 節に示すマルチエージェント逆強化学習がある．特に MA-GAIL については，交通流 [Bhattacharyya 18, Bhattacharyya 19] やエッジコンピューティング [Wang 21] といったアプリケーションへの応用も報告されているものの，群衆 Agent-based model のような部分観測環境への適用は難しい．

そこで，以下では，エージェントの観測能力が部分的な場合の関連研究として，データ駆動型 Agent-Based Model と，マルチエージェント Behavior Cloning についてまとめる．

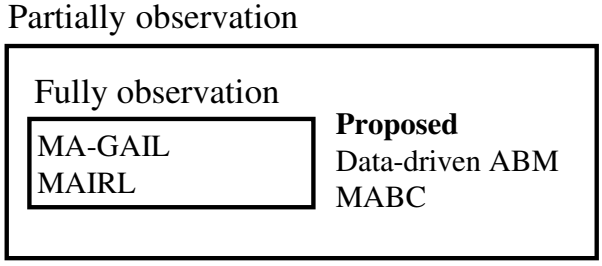


図 4.1: 行動則を推定する手法の分類（MA-GAIL：Multi-Agent Generative Adversarial Imitation Learning，MAIRL：Multi-Agent Inverse Reinforcement Learning，ABM：Agent-Based Model，MABC：Multi-Agent Behavior Cloning）

[1]観測した状態と全エージェントの行動について，状態遷移確率がマルコフ性を満たす場合

4.1.1 データ駆動型 Agent-Based Model

データ駆動型 Agent-Based Model（DABM : Data-driven ABM）は，エージェントの振舞いを記録したデータを ABM に組み込むアプローチの総称である．文献によっては，データをエージェントの初期値に用いる場合 [Sajjad 16] や，予め与えた行動則のパラメータ調整に用いる場合 [Jensen 17]，エージェントの生成から ABM の実施までの枠組みを指す場合 [Kavak 18, Truong 16] があるが，本論文ではエージェントの観測に対する行動 a を返す行動則の設計にのみ着目する．

表 4.1 に関連する DABM の文献と本提案の立ち位置を示す．DABM は，事例ベース，行動則推定，評価関数推定の三つに大別できる．

表 4.1: データ駆動型 Agent-Based Model における提案法の立ち位置（SA：Simulated Annealing，GA：Genetic Algorithm，LR：Logistic Regression，GEP：Gene Expression Programming，ADG：Automatically Defined Groups）

分類	文献	対象問題	推定法	異種性
事例ベース	[Melnikov 16]	交通流	-	異種
	[Lerner 07]	歩行者	-	同種
行動則推定	[Keller 19]	Mobile agent	GA	同種
	[Zhang 15]	Solar adoption	LR	同種
	[Zhong 15]	群衆	k-means	同種
評価関数推定	[Yamaguchi 11]	歩行者	Simplex	同種
	[Zhong 14]	群衆	GEP	同種
	提案法	群衆	ADG	異種

事例ベースは，エージェントの観測・行動対をデータベースに蓄積し，意思決定時にはそのデータベースに基づき行動則を決定する．[Melnikov 16] は，アムステルダムにおける自動車の交通流を再現している．自動車の行動則は，一日の移動ログが蓄積されたデータベースからランダム選択されたものに従う．[Lerner 07] は，歩行者の振舞いを再現するため，事例として，他エージェントとの相対位置と，そのときエージェントの選択した行動を軌跡から抽出している．エージェントの意思決定時には，観測が最も近い事例に従う．

行動則推定は，エージェントの観測から行動を出力する関数を直接推定する．[Keller 19] は，Mobile agent の行動則として，環境の観測情報から速度ベクトルを計算する関数を推定する．この関数は，予め設計された関数の組合せからなり，その組合せを遺伝的アルゴリズムを用いて求めている．[Zhang 15] は，エージェントの観測・行動の系列からロジスティック回帰により観測に対する行動をとる確率を求めている．[Zhong 15] は，群衆モデルの構築法として，目標座標ごとに k-means 法によって軌跡を分類し，各クラスタごとに二次元座標上の速度ベクトル場を求めることで，座標ごとにとる速度を推定している．

評価関数推定は，観測や行動に対する評価関数を推定し，意思決定時はその関数が最大と

なる行動を選択する．[Yamaguchi 11] は，歩行者の振舞いを再現するため，速度の変化量や衝突可能性など，予め設計された特徴量の線形和で評価関数を表し，その重みを推定する．重み推定にはシンプレックス法が用いられ，与えられた軌跡に最も近いパラメータを探索する．[Zhong 14] は，評価関数の推定を関数同定問題として定式化し，この関数同定問題を遺伝的プログラミングの拡張アルゴリズム Gene Expression Programming(GEP)[Ferreira 01] により推定している．

本論文の提案法は，群衆を対象に観測と目標状態に対する評価関数を推定する．一方，既存の DABM は，[Melnikov 16] を除き，いずれも同種の行動則を仮定しており，[Melnikov 16] は行動則を陽に扱わない点で，本提案と異なる．

4.1.2 マルチエージェント Behavior Cloning

マルチエージェント Behavior Cloning（MABC）は，エージェントの観測・行動の系列から，その軌跡の尤度を最大化する行動則を推定する機械学習法である．

既存の MABC は二つある．[Le 17] は，協調タスクにおける行動則の模倣学習を提案している．Le らは，環境の状態に併せてエージェントの役割が動的に変化することを前提に，行動則の学習に用いる軌跡を環境の状態に併せて自動的に割り当てる．[Zhan 19] は，隠れ変数を含むリカレントニューラルネットワークで方策を表現し，エージェントの目標座標を macro-intent として明示的に扱うことで協調行動の模倣精度を高めている．

本論文の提案法は，異種・同種の戦略が混在することを仮定しており，戦略の推定と同時に，戦略が同じエージェントのグループを推定する．一方，既存の MABC は個々に異なる行動則を推定しているうえ，行動則が深層ネットワークで表現されることから，推定結果からエージェントのグループを直接推定できない．

4.2 インセンティブ推定

次に，インセンティブ推定について関連研究をまとめる．まず，一般的なインセンティブ設計問題としてプリンシパル - エージェント問題を示し，本研究の用いる逆強化学習との違いを述べる．次に，逆強化学習をマルコフゲームへ拡張したマルチエージェント逆強化学習の既存法をまとめる．

4.2.1 プリンシパル - エージェント問題

プリンシパル - エージェント問題は，参加者二人のインセンティブ設計問題であり，次のように定義される [Ratliff 19].

■プリンシパル - エージェント問題：U, V をそれぞれエージェントとプリンシパルの行動，$J_p : U \times V \to \mathbb{R}, J_a : U \times V \to \mathbb{R}$ をプリンシパルとエージェントの効用，$\gamma : U \to V$

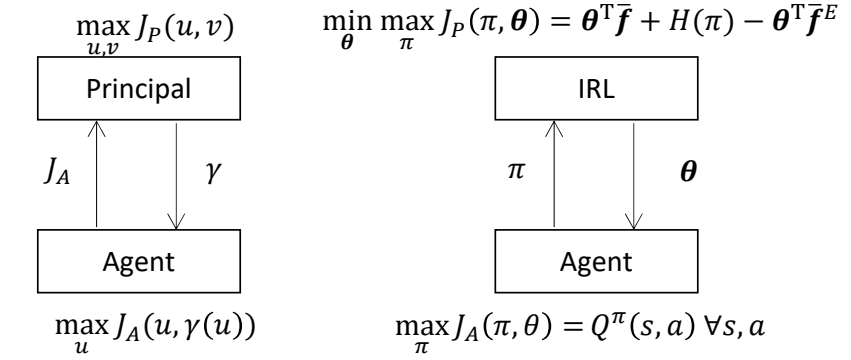


図 4.2: インセンティブ設計問題と逆強化学習の対応

をインセンティブ，$(u^d, v^d) \in \arg\max J_p(u,v)$ としたとき，$u^d = \arg\max J_a(u, \gamma(u))$ のもとで $\gamma(u^d) = v^d$ を満たす γ を見つける．

図 4.2 にインセンティブ設計問題と逆強化学習の対比を示す．逆強化学習をプリンシパル - エージェント問題としてみなす場合，逆強化学習は，逆強化学習アルゴリズムの行動 U を報酬関数の重み $\boldsymbol{\theta}$，エージェントの行動 V を最適方策 π として，与えられた軌跡とエージェントの軌跡の差を最小化していると言える．ただし，一般的なプリンシパル - エージェント問題では，エージェントの効用 J_A はプリンシパルから与えられるインセンティブに依存しない項が含まれるが，本論文の対象とするマルチエージェント逆強化学習は，与えられた報酬がエージェントの効用と一致する点で異なる．エージェントの効用に報酬以外の項を含む場合は [Wu 19] や [Mguni 19] で議論されている．

4.2.2 マルチエージェント逆強化学習

	Swarm system	Zero-sum game	Markov game
No known	[Šošić+17]	[Wang+18]	**Proposed** CD [Ziebart+10,Yang+20] AIRL [Yu+19]
Known expert reward at interaction state			[Bogert+14]
Known transition prob.		[Lin+18]	[Reddy+12]

図 4.3: マルチエージェント逆強化学習の分類（横軸は対象問題，縦軸は報酬や状態遷移確率に対する知識の有無を表す.）

図 4.3 に，マルチエージェント逆強化学習 (MAIRL : Multi-Agent Inverse Reinforcement Learning) の分類を示す．MAIRL は，対象問題と状態遷移や報酬に関する知識の有無で分

類できる[2].

Swarm system を対象とする MAIRL には，Šošić らによる推定法 [Šošić 17] がある．Swarm system とは，二つの性質（Homogeneity と Locality）をもつマルチエージェント系であり，Homogeneity は，全エージェントの観測能力や方策が同一であり，任意のエージェントが交換可能なことを意味し，Locality は，エージェントの観測が部分的であることを意味する．Šošić らは，二つの性質により全てのエージェントを同一視することで，シングルエージェント系の逆強化学習をそのまま適用できることを示した．

2 人ゼロサムゲームを対象とする MAIRL は二つある．Lin らは [Lin 18]，報酬の事前分布を仮定できるベイズの定理に基づく定式化を提案した．この提案法では，状態遷移確率所与のもと，各エージェントの方策から報酬を推定する．Wang らは [Wang 18]，与えられた軌跡と mini-max 解について，それぞれの状態価値を計算し，価値の差を最小化する報酬を獲得する枠組みを提案した．Lin ら [Lin 18] と比較した場合，状態遷移確率が不要で，準最適な軌跡が含まれていても適用可能な点が異なる．

MAIRL の中で最も一般的な Markov game を対象とする MAIRL は四つある．Reddy らは [Reddy 12]，Markov game を対象に，状態遷移確率所与のもと線形計画問題への定式化を提案した．Bogert らは [Bogert 14]，Mobile robot の振舞い予測において，エージェントが相互作用する状態における報酬既知のもと，最大エントロピー原理に基づいた推定の枠組みを提案した．Yu らは [Yu 19]，敵対的学習の枠組みを用いることで Inner Loop のない MAIRL を提案した．報酬更新毎に最適方策を求める必要がない一方で，マルチエージェント強化学習問題を内包しているため，同時学習問題が生じる課題がある．一方，文献 [Ziebart 10b, Yang 20b] は，一度に推定するエージェントを一体に限定することで，同時学習問題を回避している．

提案法は，文献 [Ziebart 10b, Yang 20b] の拡張にあたり，(1) 並列化による学習速度改善，(2)EM アルゴリズムにより報酬が同一のグループを報酬と同時に推定する．

[2]特殊な対象問題を扱う MAIRL として，全エージェントを制御する中央制御問題を対象とする場合 [Natarajan 10] や，ミクロなエージェントの振舞いではなく，マクロなエージェントの振舞いを扱う MAIRL[Yang 18] は除外した．

第5章　群衆Agent-Based Modelにおけるエージェントの異種戦略推定

本章の目的は，Agent-based model において現実の振舞いを再現するエージェントの行動則を自動推定することにある．そこで，行動則の具体例として「エージェントがどの状態を目標とするのか」に関する決定基準である戦略の推定法を提案する．提案法は，エージェント間で同種な戦略を前提とする既存の推定法 [Zhong 14] に対し,「エージェントによって戦略が異なる」異種の戦略を推定するため，Automatically Defined Groups[原 00, Hara 99] を導入する点に特徴がある．計算機実験では，避難時における群衆シミュレーションの戦略推定を例に，エージェント 20 体の環境で与えられた軌跡を再現する戦略が推定できることを示す.

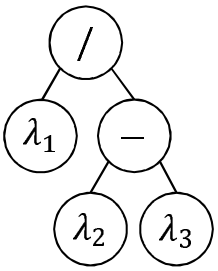
以下，5.1 節では既存法 [Zhong 14] についてまとめる．5.2 節では提案法の遺伝子表現と遺伝子操作についてそれぞれ説明する．5.3 節では提案法と既存法 [Zhong 14] を計算機実験により比較し，5.4 節で実験の結果を考察する．最後に 5.5 節で本章をまとめる．

5.1　既存法：Gene Expression Programmingによる同種戦略の推定

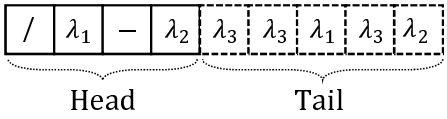
5.1.1　遺伝的プログラミングの適用

戦略の推定を，軌跡 $\mathcal{M}^*$，特徴量集合 $\mathcal{T}$，関数集合 $\mathcal{F}$ を所与として，$\mathcal{M}^*$ と $\hat{\mathcal{M}}$ の誤差を最小化する戦略 ϕ を見つける関数同定問題とする．$\mathcal{M}^*$，$\hat{\mathcal{M}}$ はいずれも全エージェントの軌跡であり，軌跡はエージェントの座標系列を意味する．$\mathcal{M}^*$ は再現対象の軌跡，$\hat{\mathcal{M}}$ は戦略 ϕ でシミュレーションした軌跡をそれぞれ表す．また戦略 ϕ は，目標状態 g の間で共通して観測される特徴量集合 $\mathcal{T} = \{\lambda_k \in \mathbb{R} \mid k \leq K, k \in \mathbb{N}\}$，および複数のスカラー量から一つのスカラー量を計算する関数の集合 $\mathcal{F}$ の各要素の組み合わせからなる．ただし，戦略はエージェント間で共通し，エージェント i が目標状態 g に対して実際に観測する特徴量の値 λ_{igk} はエージェントや目標状態によって異なる．

以下では戦略 ϕ を，関数を親，その引数を子としてラベル付けした木構造として扱う．この変換により戦略 ϕ の推定は，木の組合せ最適化問題として遺伝的プログラミングにより解くことができる．図 5.1(a) に $\phi = \lambda_1/(\lambda_2 - \lambda_3)$ を木構造で表した例を示す．この戦略に従う場合，λ_1, λ_3 の値が大きく，かつ λ_2 の値が小さい目標状態が選ばれる．



(a) GEP の木構造



(b) GEP の遺伝子表現

図 5.1: GEP における木構造と遺伝子表現の例

5.1.2 アルゴリズム設計

Zhong ら [Zhong 14] は，GEP[Ferreira 01] を用いて関数同定問題を解いている．GEP は，2.1 節の遺伝的アルゴリズムと 2.2 節の遺伝的プログラミングを組み合わせた進化計算の手法である．GEP の遺伝子表現は，木構造を長さ一定の文字列として扱う．そのため，有限な木構造の組み合わせを，遺伝的アルゴリズムと同様の遺伝的オペレータを用いて探索する．

遺伝子表現

図 5.1(b) に図 5.1(a) の遺伝子表現の例を示す．各個体は Head および Tail から構成され，Head は特徴量集合 $\mathcal{T}$ または関数集合 $\mathcal{F}$ の要素，Tail は $\mathcal{T}$ の要素からなる[1]．ただし，木構造を維持するため，Head および Tail は $L_{\text{tail}} = L_{\text{head}} \cdot (\arg - 1) + 1$ を満たす必要がある[2]．ここで，L_{head} は Head の長さ，L_{tail} は Tail の長さ，arg は $\mathcal{F}$ のうち最大の引数の数を表す．

遺伝子表現から木構造への写像は全射である．木構造で表される戦略を配列の遺伝子表現へ変換する場合，木構造のノードを幅優先探索の順で走査し，そのラベルを配列上に並べる．一方，遺伝子表現から戦略を復元する場合は，配列を先頭から順に走査し，その要素をノード，要素が関数集合に含まれる場合は，引数の数のリンクを生成し，幅優先探索の順に構築する．そのため，例えば，遺伝子表現の Head に特徴量集合の要素が含まれる場合，配列上の要素のうち木構造上に現れない要素が生じる．

遺伝的オペレータ

個体に対して，適応度に応じて選択，突然変異，転移，交叉する．選択はバイナリトーナメント選択，交叉は一点交叉および二点交叉を用いる．

[1]本章の特徴量集合および関数集合は，2.2 節の終端記号集合および非終端記号集合に一致する．

[2]Head が全て関数集合の要素からなる場合，木構造上では L_{head} 個のノードと $L_{\text{head}} * \arg$ 本のエッジが生じ，各エッジは頂点以外の非終端ノードか，終端ノードに張られる．つまり，終端ノード，つまり特徴量集合の要素がラベル付けされたノードへの接続は $L_{\text{head}} * \arg - L_{\text{head}} + 1$ 本ある．よって，Head の長さに対し，Tail の長さが $L_{\text{tail}} = L_{\text{head}} \cdot (\arg - 1) + 1$ を満たせば，任意の遺伝子表現を木構造に変換できる．

適応度評価

各個体の適応度は，式 (5.1) に示す軌跡 $\mathcal{M}^*$ と個体の表わす戦略でシミュレーションした軌跡 $\hat{\mathcal{M}}$ の推定誤差 $D(\mathcal{M}^*, \hat{\mathcal{M}})$ によって評価する．推定誤差 D は，各タイムステップ t ごとに軌跡 $\mathcal{M}$ の観測値 $\{\xi_{\mathcal{M}}(p,t) \mid p \leq P, p \in \mathbb{N}\}$ を比較して評価する．また，T は総タイムステップ数である．

$$D(\mathcal{M}^*, \hat{\mathcal{M}}) = \frac{\sum_{t=1}^{T} \sum_{p=1}^{P} |\xi_{\mathcal{M}^*}(p,t) - \xi_{\hat{\mathcal{M}}}(p,t)|}{TP} \tag{5.1}$$

5.2 提案法：Automatically Defined Groups を導入した異種戦略の推定

5.2.1 概要

3.1 節で述べたように，現実の群衆には，同じ戦略を共有する同種のエージェントと，異なる戦略を持つ異種のエージェントが混在することから，同じ戦略に従うエージェントはグループ化しつつ，各グループごとに異なる戦略を推定する方法が必要となる．これに対して本節では，Automatically Defined Groups（ADG）[Hara 99] の枠組み，および赤池情報基準量を導入した異種戦略の推定法を提案する．

ADG は，マルチエージェント系において，タスク達成のための行動則を学習するチーム学習 (Team Learning) の一つである [Panait 05]．具体的には，同じ行動則に従うエージェントのグループと，その行動則を最適化する遺伝的プログラミングの拡張法であり，医療診断データの解析 [Hara 05] や，マルチエージェント型の人工株式市場の構築 [Ogino 04] に用いられている．そこで，ADG における行動則を戦略として扱い異種の戦略を推定する．

また，5.1.2 節で述べた通り，GEP は有限な木構造を探索するため，ブロートの抑制だけでなく，推定される戦略の単純化によって，可読性を高めることが期待できる．一方で，ADG は木構造に制限がないため，ノード数の増加により，推定される戦略の可読性を低下させる可能性がある．そこで，GP における木構造のもつ複雑さの計算法と，その値に基づくモデル選択法 [Le 16] のうち，本提案では赤池情報基準量 (AIC) を用いた手法 [Borges 10] を参考に適応度関数を設定する．

5.2.2 アルゴリズム設計

ADG は木構造の探索に加え，エージェントのグループ構成も同時に探索するため，一般的な遺伝的プログラミングとは遺伝子表現だけでなく，遺伝的オペレータも異なる．以下では，ADG を導入した遺伝子表現および遺伝的オペレータについてそれぞれ説明し，赤池情報基準量を導入した適応度評価について述べる．

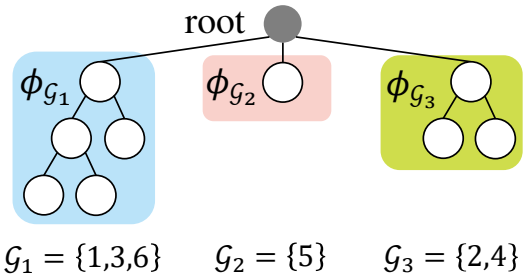


図 5.2: 提案法の遺伝子表現例

遺伝子表現

図 5.2 にエージェント 6 体，グループ数 3 の遺伝子表現の例を示す．グループ $\mathcal{G}$ をエージェント集合，$\phi_{\mathcal{G}}$ を $\mathcal{G}$ に含まれるエージェントが従う戦略とする．各個体は，図 5.2 中の root が全グループの戦略を子としてもつ一つの木構造と，各グループの構成 $\mathcal{G}$ を含むデータ構造をもつ．ただし，一般的な遺伝的プログラミングと同様に，$\phi_{\mathcal{G}}$ の深さには制限を設けない．初期個体は，エージェントをランダムにグループ分けし，各グループの戦略もランダムに生成する．

遺伝的オペレータ

■交叉：交叉では，戦略の交叉に加えグループ構造も変化させる．はじめに，任意のエージェント k を選択し，2 個の親個体 α, β のうち k を含むグループ $\mathcal{G}_\alpha, \mathcal{G}_\beta$ を各々求める．つぎに，以下の三つの条件に従いグループを変化させたのち，$\phi_{\mathcal{G}_\alpha}, \phi_{\mathcal{G}_\beta}$ を交叉する．

type a $\mathcal{G}_\alpha = \mathcal{G}_\beta$ のとき各グループは変化させない．

type b $\mathcal{G}_\alpha \supset \mathcal{G}_\beta$ または $\mathcal{G}_\alpha \subset \mathcal{G}_\beta$ のとき，$\mathcal{G}_\alpha \cap \mathcal{G}_\beta$ だけが交叉の影響を受けるようにグループを分割する．

type c $\mathcal{G}_\alpha$ と $\mathcal{G}_\beta$ が包含関係にないとき，$\mathcal{G}_\alpha \cup \mathcal{G}_\beta$ が同じ戦略に従うようにグループを統合する．

図 5.3 にエージェント 2 が選択された場合の具体例を示す．図中の {} 内の数字はグループに含まれるエージェント番号 ($AgentNo.$) を表す．**type a** の場合，$\mathcal{G}_\alpha = \mathcal{G}_\beta = \{1,2\}$ であるため，グループは変化させずに交叉する．**type b** の場合，$\mathcal{G}_\beta = \{1,2,3\}$ が $\mathcal{G}_\alpha = \{2\}$ を包含するため，$\mathcal{G}_\beta$ を $\{1,3\}$ と $\{2\}$ に分割し，エージェント 2 を新たなグループに移動させたのちに交叉する．移動時，$\{2\}$ の戦略は移動元の戦略 $\phi_{\mathcal{G}_\beta}$ と同一にする．**type c** の場合，$\mathcal{G}_\alpha = \{1,2\}$ と $\mathcal{G}_\beta = \{2,3\}$ が包含関係にないため，各個体のグループを $\mathcal{G}_\alpha \cup \mathcal{G}_\beta = \{1,2,3\}$ に統合したのちに交叉する．個体 α では，$\mathcal{G}_\alpha = \{1,2\}$ に $\{3\}$ を移動し，移動元の戦略は削除する．同様に，個体 β では，$\mathcal{G}_\beta = \{2,3\}$ に $\{1\}$ を移動し，移動元の戦略は削除する．

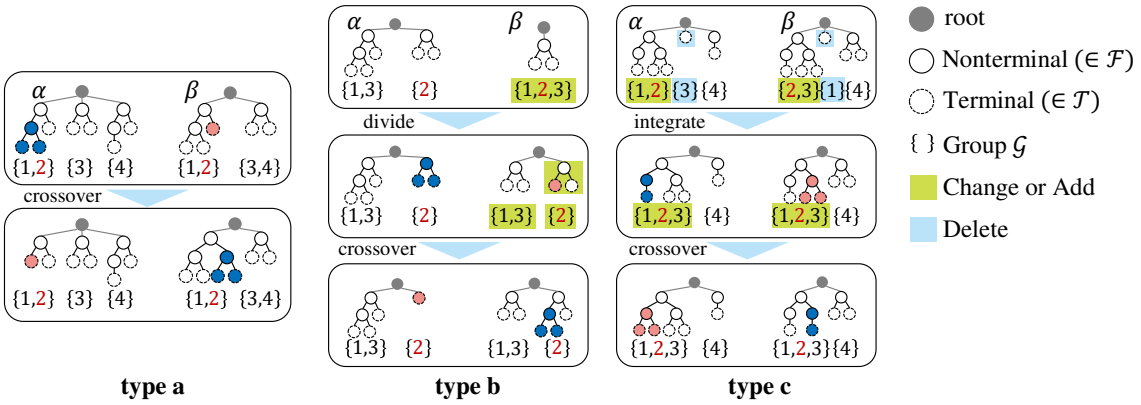


図 5.3: 交叉例

■グループ突然変異：グループ突然変異では，グループ構造の局所解への収束を抑制するため，交叉の前に，各個体のグループ構成をグループ突然変異率に従い変異させる．具体的には，任意のエージェント i を選択し，現在属するグループ $\mathcal{G}$ から任意のグループ $\mathcal{G}'$ へ移動する．ここで，$\mathcal{G}' = \mathcal{G}$ が選択された場合は，新たなグループ $\{i\}$ を作成する．このとき，$\{i\}$ の戦略は移動元の戦略 $\phi_{\mathcal{G}}$ と同一とする．

■選択：選択にはトーナメント選択を用いる．具体的には，集団からトーナメントサイズ n だけ個体をランダムに選択し，選択された個体のうち最も適応度の高い個体を選択する．ただし，最も適応度の高い個体が複数存在した場合は，文献 [Hara 99] と同様にグループ数が最小の個体を選択する．この選択基準は，グループの過度な分割を抑制することを意図している．

適応度関数

戦略の複雑さを「戦略のノード数」として計算し，ノード数が少ないほど適応度が高くなるよう適応度関数を定義する．定義には，GP に AIC を用いた文献 [Montaña 11, Le 16] を参考にした．

文献 [Montaña 11, Le 16] では，n 対の入出力データ $(x_i, y_i)_{i=1}^{n}$ に対し，GP の個体 f の複雑度を式 (5.2) で評価する．

$$\varepsilon(f) = \varepsilon_n(f) + \frac{2h}{n}\sigma^2 \tag{5.2}$$

$$\varepsilon_n(f) = \frac{1}{n}\sum_{1\le i\le n} Q(x_i, f : y_i) \tag{5.3}$$

$$\sigma^2 = \frac{n}{n-h}\frac{1}{n}\sum_{1\le i\le n}(y_i - f(x_i))^2 \tag{5.4}$$

ここで，$\varepsilon_n(f)$ はモデル f のデータに対する平均誤差，$Q(x_i, f : y_i)$ はモデル f の予測とデータの誤差，h はモデルの複雑度をそれぞれ表す．[Montaña 11] では，h に個体に含まれる変数ノードの数を用いている．

前述の AIC を本提案に導入した場合の適応度を式 (5.5)，戦略の複雑度 h を式 (5.7) に示す．簡単のため，戦略の複雑度には戦略のノード数を用いた．ここで，$\mathrm{Node}(\phi_{\mathcal{G}})$ は戦略 $\phi_{\mathcal{G}}$ のノード数，$|\mathcal{G}|$ はグループのエージェント数をそれぞれ表す．式 (5.5) の第二項目は，戦略のノード数 h に関するペナルティを表わす．

$$F = D(\mathcal{M}^*, \hat{\mathcal{M}}) + \frac{2h}{TP}\sigma^2 \tag{5.5}$$

$$\sigma^2 = \frac{1}{TP-h}\sum_{t=1}^{T}\sum_{p=1}^{P}(\xi_{\mathcal{M}^*}(p,t) - \xi_{\hat{\mathcal{M}}}(p,t))^2 \tag{5.6}$$

$$h = \sum_{\mathcal{G}} |\mathcal{G}|\ \mathrm{Node}(\phi_{\mathcal{G}}) \tag{5.7}$$

5.2.3 計算量評価

エージェント数 N に対して，1 世代あたりの実行計算量を評価する．適応度評価においては，推定された戦略を用いて群衆シミュレーションを実行し，全エージェントの軌跡 $\hat{\mathcal{M}}$ を計算する．そのため，対象とする群衆シミュレーションに依存して適応度評価の計算量が決まる．例えば，5.3 節で述べる実験に用いる群衆避難モデルは $O(N^2)$ である．一方，遺伝的オペレータによる操作は，交叉を除き，定数時間で実行される．交叉において適用する type を判定するとき，選択された二つのグループ $\mathcal{G}_\alpha, \mathcal{G}_\beta$ の包含関係を判定する．そのため，判定の実装方法によって計算量が異なる．例えば，片方のグループからエージェントを選び出し，もう片方のグループに属するか判定する素朴な方法を用いると $O(N^2)$，二分探索とクイックソートなど比較的効率のよい手法を組み合わせると $O(N \log N)$ となる．

5.3 計算機実験

本節では，避難時における群衆の戦略推定を例として，提案法と Zhong らによる既存法 [Zhong 14] を比較する．実験は，エージェントに 5.3.1 節で述べる群衆避難モデルを仮定し，方策を 2.3 節で述べた Socail Force Model(SFM)[Helbing 00] としたときの戦略を推定する．

5.3.1 群衆避難モデル

群衆避難モデルは，Zhong ら [Zhong 14] と同様に，避難する出口を選択する上位階層，および次ステップの座標を選択する下位階層によって構成する[3]．各避難者は，出口を通過して屋外へ避難するまで，出口および座標の選択を繰り返すエージェントとする．

[3] 上位階層および下位階層は 3.1 節で述べた戦略および方策にそれぞれ対応する．

■上位階層：エージェント i は，戦略 ϕ_i に基づき出口 g_i を選択する．戦略 ϕ_i は，式 (5.8) の特徴量集合 $\mathcal{T}$ および式 (5.9) の関数集合 $\mathcal{F}$ の各要素の組み合わせからなる．図 5.1 に各特徴量の値とその意味を示す.

$$\mathcal{T} = \{d, w, \eta, f\} \tag{5.8}$$

$$\mathcal{F} = \{+, -, /, *, F\} \tag{5.9}$$

表 5.1: 各特徴量の数値に対する意味

Feature λ_k	$\lambda_k = -1$	$\lambda_k = 0$
$\lambda_0 = d$	出口から遠い	出口に近い
$\lambda_1 = w$	出口幅が狭い	出口幅が広い
$\lambda_2 = \eta$	混雑	空き
$\lambda_3 = f$	非追従	追従

特徴量 d および w は，出口 g までの距離および幅をそれぞれ表す．η は，出口 g を選択かつ出口 g に近い他エージェントの数である．f は近傍エージェントに対する追従を表し，近傍エージェントが出口 g を選択しているとき 0，そうでなければ -1 を示す．簡単のため，近傍エージェントは最も距離の近いエージェントとし，視界は考慮しない．各特徴量 λ は，部屋の形状や全エージェント数に合わせて $-1 \leq \lambda \leq 0$ に正規化する．出口の選択時，δ_g 最大の出口が複数ある場合はその中からランダムに選択する．また，δ_g が全て計算不可能，または無限大に発散する場合は，全ての出口からランダムに選択する．関数集合 $\mathcal{F}$ は，四則演算と，引数 x の正負を変換する $F(x) = -x$ からなる．

■下位階層：2.3 節に示した SFM[Helbing 00] に従う．上位階層で選択された出口 g_i は，SFM の目標方向 $\mathbf{e}_i^0(t)$ をエージェント i から出口 g_i への単位ベクトルとすることで，出口へ向かう次ステップの座標を選択する．

以上から，群衆避難モデルによるシミュレーション一回あたりの実行計算量を評価する．エージェント数 N，戦略の計算コスト E，総ステップ数 T に対して，各ステップごとに特徴量 η, f の計算量は $O(N^2)$，戦略による δ_g の計算量は $O(EN)$，式 (2.1) の計算量は $O(N^2)$ となる．よって，実行計算量は $O((N^2 + EN)T)$ である．

5.3.2 実験設定

■評価データの生成

推定結果の妥当性を評価するため，群衆避難モデルに幾つかの戦略を与え，3 種類の評価データを生成した．表 5.2 に各評価データに設定した戦略を示す．実験では，避難するエージェントは 20 体とした．一つめは，同種の戦略で振舞う評価データ [d],[w] である．このデー

表 5.2: 評価データの戦略 (n はエージェントのインデックス．特徴量 d, w, η, f は表 5.1 に示した意味を持つ．つまり，各戦略 ϕ は，d のとき最短距離の出口，w のとき幅最大の出口，f のとき近傍エージェントへの追従，$d+\eta$ のとき混雑度最小かつ最短距離の出口を選択する．また，Manual は手動操作による軌跡生成を意味する．)

Evaluation Data	Strategy $\phi_{\{\text{AgentNo.}\}}$
[d]	$\phi_{\{0\leq n<20\}} = d$
[w]	$\phi_{\{0\leq n<20\}} = w$
[d,w]	$\phi_{\{2n\|0\leq n<10\}} = d$, $\phi_{\{2n+1\|0\leq n<10\}} = w$
[d,f]	$\phi_{\{0\leq n<15\}} = d$, $\phi_{\{15\leq n<20\}} = f$
[w,f]	$\phi_{\{0\leq n<15\}} = w$, $\phi_{\{15\leq n<20\}} = f$
[d,η]	$\phi_{\{0\leq n<15\}} = d$, $\phi_{\{15\leq n<20\}} = d+\eta$
[w,η]	$\phi_{\{0\leq n<15\}} = w$, $\phi_{\{15\leq n<20\}} = d+\eta$
[M,d,f]	$\phi_{\{0\}} = \text{Manual}$, $\phi_{\{1\leq n<15\}} = d$, $\phi_{\{15\leq n<20\}} = f$

タは，エージェントの観測する特徴量が全て既知，かつデータにノイズを含まない状況で戦略を推定する．二つめは，異種の戦略で振舞う評価データ [d,w]〜[w,η] である．このデータについても，特徴量が全て既知，かつデータにノイズを含まない状況で戦略を推定する．三つめは，異種の戦略で振舞う評価データ [M,d,f] である．このデータは [d,f] のうちエージェント 0 を手動操作して生成した．そのため，エージェントの観測する特徴量が完全には分からず，かつデータにノイズを含む状況で戦略を推定する．

図 5.4 に実験対象とした仮想の部屋と，評価データ [M,d,f] の軌跡を示す．赤の太線で示した軌跡は，手動操作したエージェント 0 の軌跡を表わす．文献 [Zhong 14] と同様に，部屋には幅の異なる四つの出口があり，そのいずれかから避難する．図 5.5 に実験で用いたエージェントの初期位置を全て示す．エージェントの初期位置は，[M,d,f] は図 5.5(c) の 1 種類，それ以外の評価データでは図 5.5 の全 10 種類を用いた．よって，今回実験に用いた軌跡は計 71 種類である．

■適応度関数

式 (5.1) で表される適応度関数の観測値 ξ には，[Zhong 14] と同様に，式 (5.10) のタイムステップ t における座標 x のエージェントの局所密度 [Helbing 07] を用いる．ここで，N は全エージェント数，$r_i(t)$ はタイムステップ t におけるエージェント i の座標，R はパラメータである．

$$\rho(x,t) = \sum_{i=1}^{N} \frac{1}{\pi R^2} \exp[-||r_i(t) - x||^2 / R^2] \tag{5.10}$$

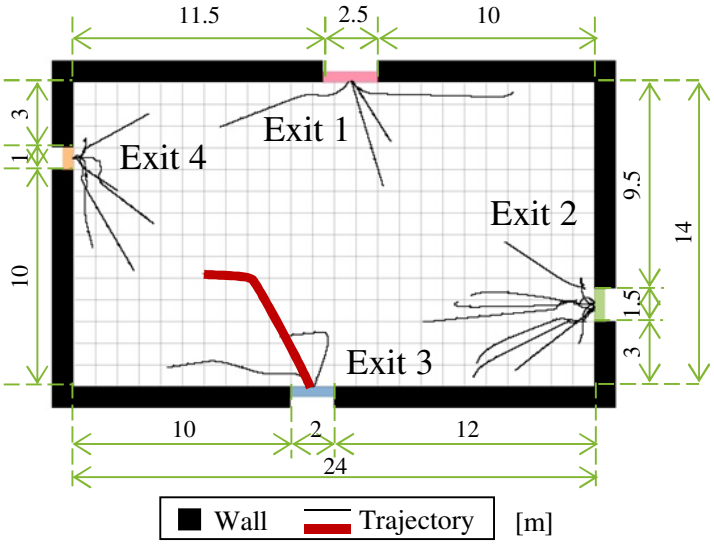


図 5.4: 部屋環境と評価データ [M,d,f] の軌跡

■推定精度の評価基準

推定精度は二つに指標により評価する．

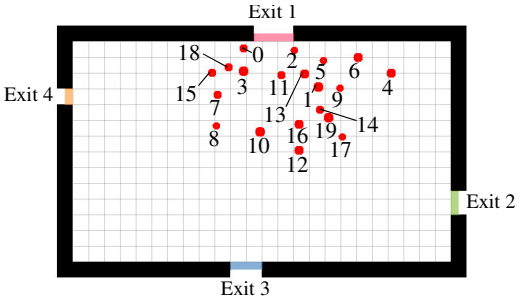
一つめは，適応度関数に用いる推定誤差 D であり，軌跡全体の誤差を評価する．推定誤差は，式 (5.10) から 0.1[s] 間隔でエージェントの密度を計算し，式 (5.1) から 1 グリッドにおける 0.1[s] あたりの密度差の平均値とした．式 (5.10) のエージェントの密度は，図 5.4 に示すグリッドごとに評価した．各グリッドは，1[m] 間隔で縦に 14，横に 24 分割した．また，総ステップ数 T は各評価データで全エージェントが避難完了したステップとした．

二つめは，出口の通過誤差 D_p であり，各出口における通過タイミングと人数の誤差を評価する．出口の通過誤差 D_p は，出口 g においてステップ t における累積通過人数 $n_{e,t}$ に対して式 (5.11) で評価する．ここで，Exit は全出口数を表わす．

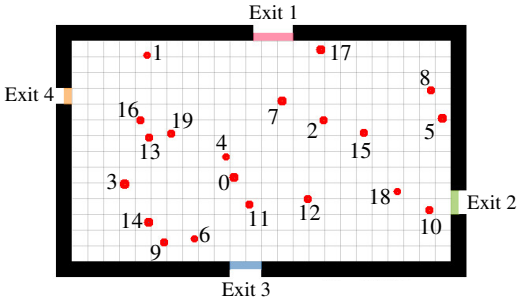
$$D_p = \frac{1}{|\mathrm{Exit}|} \sum_{g \in Exit} \sqrt{\frac{1}{T} \sum_{0 \le t \le T} (n_{g,t}^{\mathcal{M}^*} - n_{g,t}^{\hat{\mathcal{M}}})^2} \tag{5.11}$$

■パラメータ

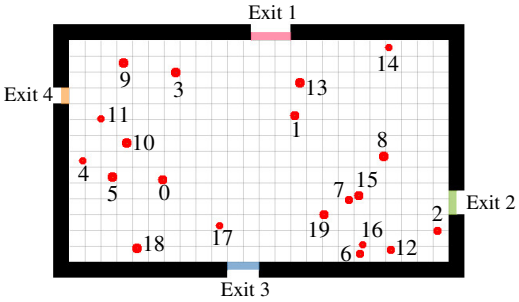
提案法のパラメータは文献 [Hara 99] から，トーナメントサイズ 5，突然変異率 0.9，グループ突然変異率 0.01，交叉率 0.9 とした．また，個体数を 200 とした．一方，既存法 [Zhong 14] のパラメータは文献 [Zhong 14] から，遺伝子長 13，突然変異率 0.1，転移率 0.1，交叉率 0.7，個体数 10 とした．また，どちらの手法も最大世代数を 200 とした．



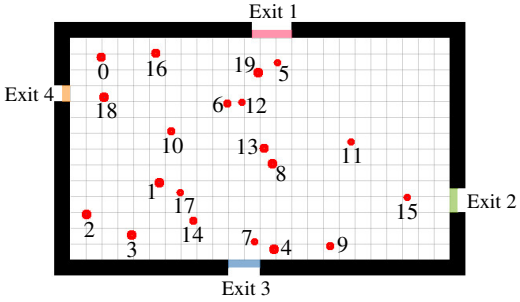
(a) Pos. 1



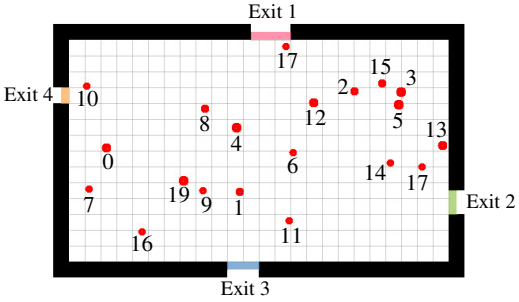
(b) Pos. 2



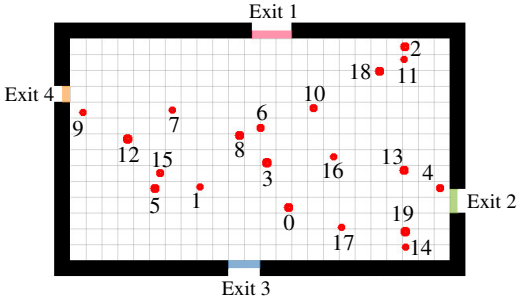
(c) Pos. 3



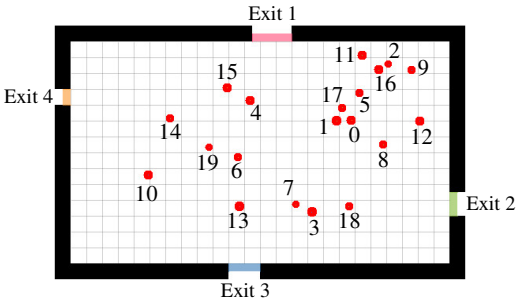
(d) Pos. 4



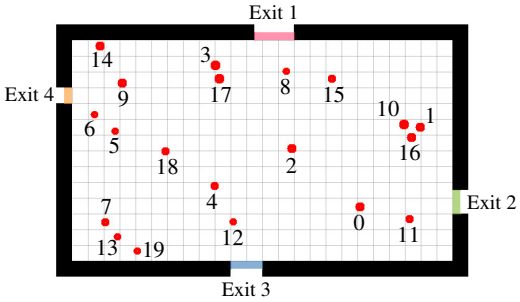
(e) Pos. 5



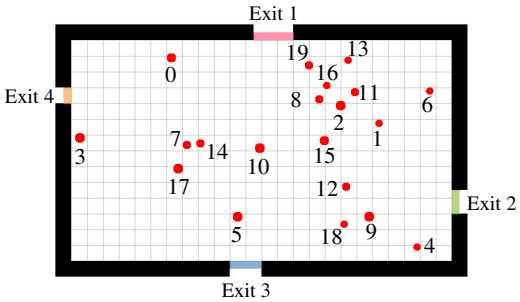
(f) Pos. 6



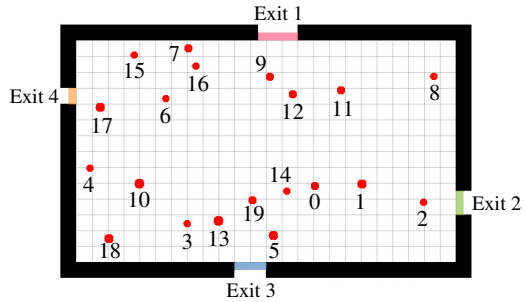
(g) Pos. 7



(h) Pos. 8



(i) Pos. 9



(j) Pos. 10

図 5.5: エージェントの初期配置（図中の数字は $AgentNo.$ を表す.）

表 5.3: 推定成功率 [%] の比較

	[d]	[w]	[d,w]	[d,f]	[w,f]	[d,η]	[w,η]	[M,d,f]
Previous	**100**	**100**	10	0	0	43	0	0
Proposed w/o AIC	**100**	**100**	**75**	13	27	**77**	43	0
Proposed	**100**	**100**	70	**18**	**31**	76	**47**	0

■比較方法

実験では，提案法との比較対象として Zhong らの既存法 [Zhong 14] を取り上げる．GEP[Ferreira 01] の拡張版として Self-learning gene expression programming[Zhong 16] などが提案されているが，主な拡張版は全て探索の正確性または効率性の改善を目的としている [Zhong 16]．そのため，これらの拡張版も 5.2.1 節で述べた複数の戦略を表現できない．したがって，本章の趣旨である戦略の推定法として比較対象は GEP を用いた既存法 [Zhong 14] で十分と判断した．

また，提案法に導入した AIC の効果を比較するため，式 (5.1) を適応度関数とした場合も比較した．各手法は，計 71 種類の各評価データに対して，10 試行中の推定成功率およびその推定精度に基づき比較する．本章は評価データと一致する戦略推定を目的とするため，式 (5.1) の推定誤差 $D = 0$ の個体が得られたとき，推定に成功したとみなす．

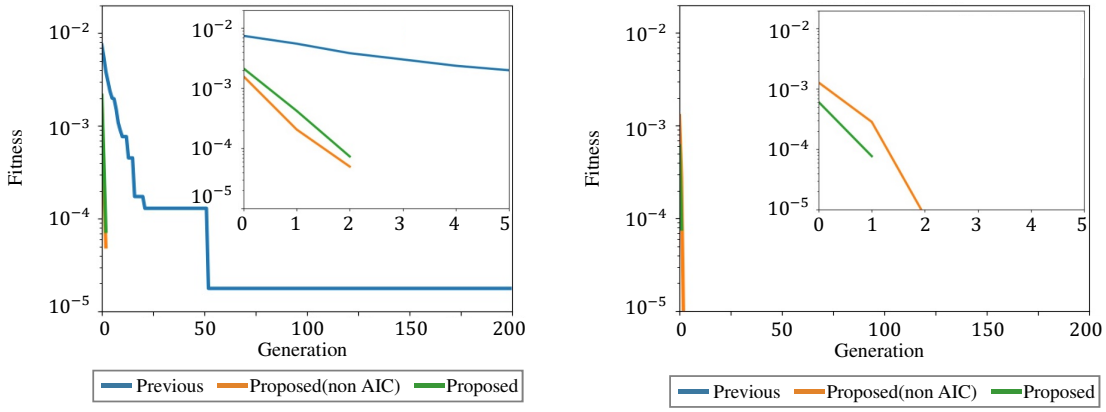
5.3.3 実験結果

推定成功率および推定精度

まず，表 5.3 に評価データごとの推定成功率，表 5.4 に解の推定誤差 D の平均値，および標準偏差をそれぞれ示す．この結果から，評価データ [d],[w] の場合，既存法，提案法ともに，全試行で推定に成功していることがわかる．一方，[M,d,f] を除く他の評価データの場合，既存法と比較して提案法の成功率が高く，解の推定誤差 D は全て提案法の方が小さいことがわかる．また，手動操作を含む評価データ [M,d,f] の場合は，既存法，提案法ともに，全試行で推定に失敗したが，解の推定誤差 D は提案法の方が小さいことがわかる．

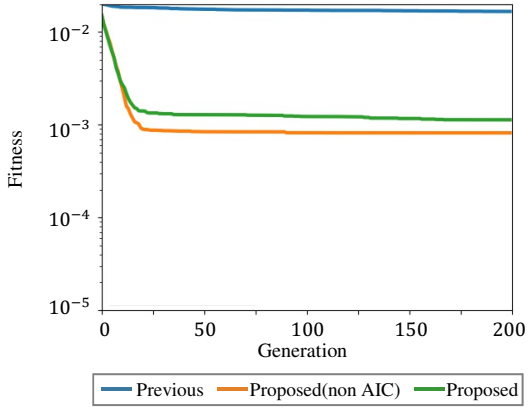
つぎに，図 5.6，図 5.7 に各世代における最適個体の適応度を各評価データごとに平均した結果を示す．各図から，どの手法においても適応度は最適解または局所解に収束していることがわかる．

最後に，評価データ [d,w]〜[M,d,f] のうち，推定に失敗した場合の推定精度を確かめた．図 5.8 に，出口の通過誤差 D_p と，解の推定誤差 D のプロットを示す．図 5.8 には 10 試行のうち推定誤差 D 最小の解だけをプロットしている．プロットの分布から，提案法は既存法と比較して，D，D_p ともに推定精度が高い解が多く含まれることがわかる．また，評価データ [M,d,f] の推定精度についても同様に，提案法の方が精度が高い．

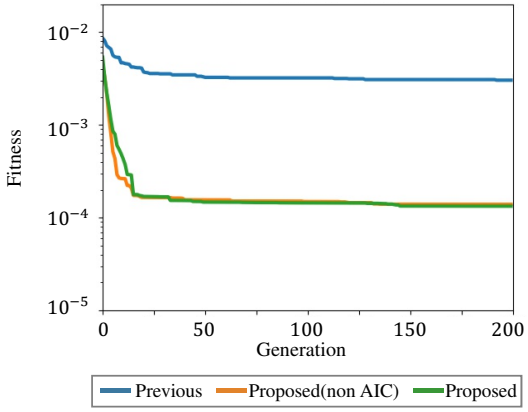


(a) 評価データ [d]

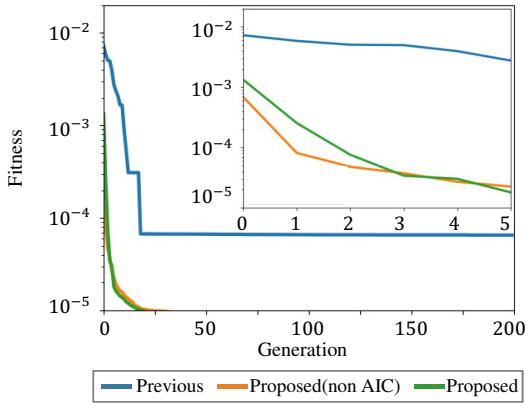
(b) 評価データ [w]



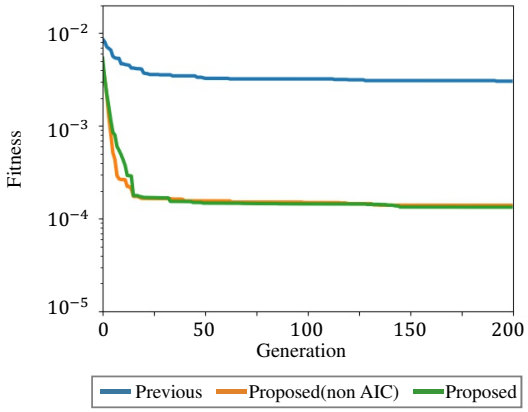
(c) 評価データ [d,w]



(d) 評価データ [d,f]



(e) 評価データ [w,f]

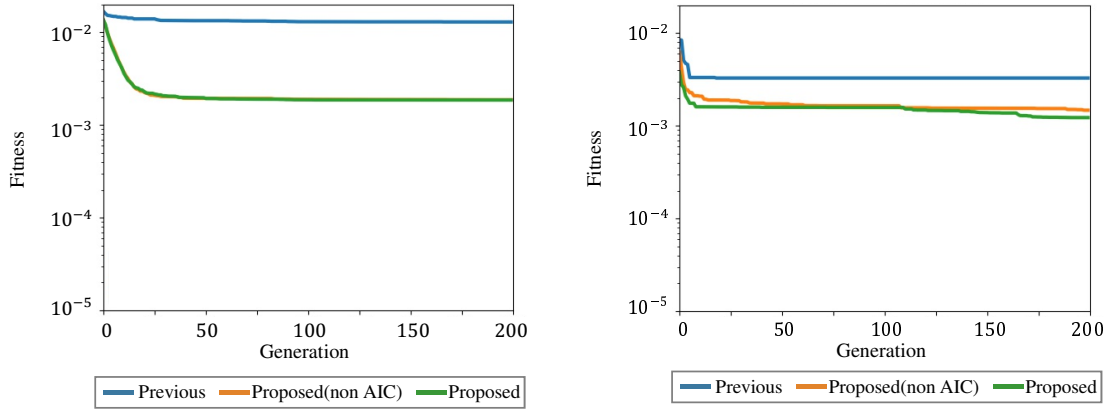


(f) 評価データ [d,η]

図 5.6: 最良個体に対する適応度の平均値推移（縦は対数軸のため，適応度が 0 になったときプロットが途切れる.）

表 5.4: 推定誤差の平均と標準偏差の比較

	[d]	[w]	[d,w]	[d,f]	[w,f]	[d,η]	[w,η]	[M,d,f]
Previous	**0.0**	**0.0**	1.7E-2	3.1E-3	6.5E-5	3.4E-3	1.3E-2	3.4E-3
	±0.0	±0.0	±7.1E-3	±3.1E-3	±8.7E-5	±5.4E-3	±3.0E-3	±0.0
Proposed w/o AIC	**0.0**	**0.0**	**8.2E-4**	1.4E-4	8.3E-6	7.0E-4	1.9E-3	1.6E-3
	±0.0	±0.0	±1.7E-3	±5.6E-4	±1.1E-5	±2.1E-3	±2.3E-3	±6.5E-4
Proposed	**0.0**	**0.0**	9.2E-4	**1.1E-4**	**5.9E-6**	**4.3E-4**	**1.7E-3**	**1.2E-3**
	±0.0	±0.0	±1.5E-3	±4.8E-4	±8.2E-6	±1.5E-3	±2.1E-3	±4.7E-4



(a) 評価データ [w,η]　　(b) 評価データ [M,d,f]

図 5.7: 最良個体に対する適応度の平均値推移（縦は対数軸のため，適応度が 0 になったときプロットが途切れる.）

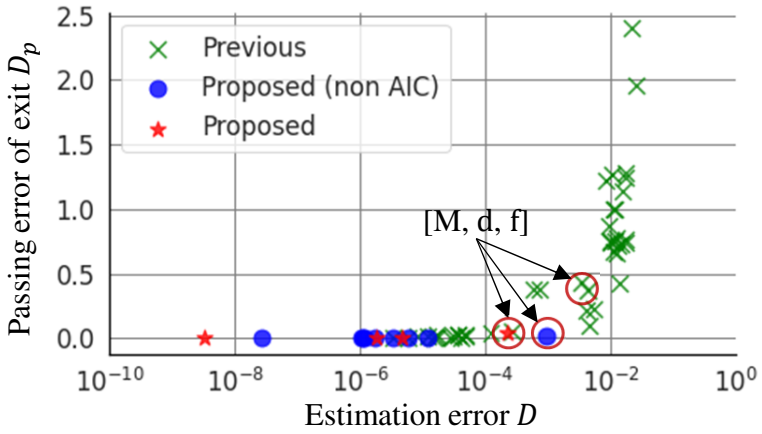


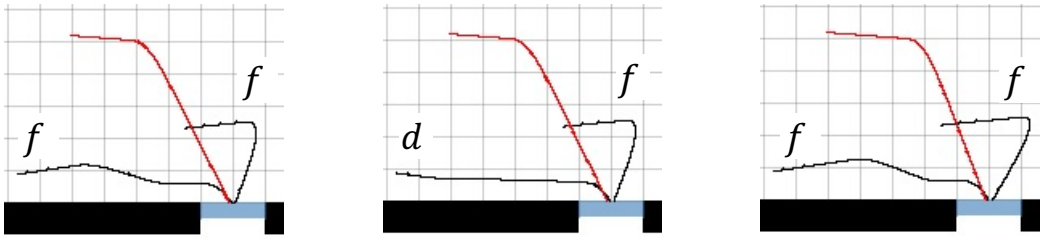
図 5.8: 推定結果の出口の通過誤差 D（縦軸）と累積避難者数の平均二乗誤差 D_p（横軸）（各プロットが最良個体の値を示す.）

戦略の推定結果

表 5.5: 提案法による推定例

Evaluation Data	Estimated Strategy $\phi_{\mathcal{G}}$
[d]	$\phi_{\text{all}} = d$ $\phi_{\text{all}} = w$ $\phi_{\text{all}} = d + w$ $\phi_{\text{all}} = (d+1)w$ $\{\phi_{2,7,8,15,19} = d, \phi_{\text{other}} = w\}$
[d,w]	$\phi_{\text{all}} = d$ $\phi_{\text{all}} = w$ $\phi_{\text{all}} = \eta + (1-f)(d+w)$ $\{\phi_{\{16\}} = d^2/f - f - 1, \phi_{\{13\}} = w + \eta,$ $\phi_{\{1,2,5,9,3,11,15,17,19\}} = w, \phi_{\text{other}} = d\}$
[M,d,f]	$\{\phi_{\{0\}} = w - \dfrac{\eta}{\dfrac{\eta}{2df + wf + \eta} - 2w + d + \eta - f},$ $\phi_{\{4,10,13,17,19\}} = f, \phi_{\text{other}} = d\}$

表 5.5 に提案法による戦略の推定例を示す．評価データ [d],[d,w] は推定誤差 $D = 0$ の結果例，評価データ [M,d,f] は 10 試行のうち適応度最小の結果をそれぞれ示している．評価データ [d],[d,w] の結果から，表 5.2 で定めた戦略と異なる結果も推定されたことがわかる．例えば，評価データ [d] では最短距離避難の戦略 d を全エージェントに設定した．その推定結果には，d だけでなく，一部または全エージェントに，出口までの幅 w をもつ戦略も含まれている．



(a) (b) (c)

図 5.9: 評価データ [M,d,f] における手動操作したエージェント 0（赤線）付近の軌跡比較（(a) は評価データの軌跡，(b) は推定結果の軌跡（推定誤差 $D = 2.3 \cdot 10^{-4}$），(c) はエージェント 0 は推定結果，その他は評価データの戦略に従った場合の軌跡（推定誤差 $D = 7.4 \cdot 10^{-4}$）をそれぞれ表す.）

また，評価データ [M,d,f] については，手動操作したエージェント 0 に対して，4 つの特

表 5.6: エージェント 17 に対する戦略の推定結果（複数の試行結果のうち重複する戦略は除いた.）

Evaluation Data	True	Estimated Strategy of	
		Pos.1	Pos.2
[d]	d	d,w,$(d+1)w$, $d+w$	d
[w]	w	d,w	w
[d,f]	f	$d-\eta(\eta+f-2d+1)$	f
[w,η]	$d+\eta$	$\eta, 3d+\eta+f+\frac{f\eta}{w}-\frac{d\eta}{w}-\frac{\eta^2}{w}+1$	d,w

徴量を全て含む非線形関数が推定され，それ以外の戦略は表 5.2 と一部異なる．手動操作したエージェント 0 については，正しい戦略がわからないため，その妥当性評価が難しい．そこで，推定された戦略に従った場合の軌跡によって，その妥当性を確認する．図 5.9 にその結果を示す．図 5.9(a) は評価データの軌跡，図 5.9(b) は全ての推定結果に従った軌跡，図 5.9(c) はエージェント 0 の推定結果以外は評価データと同じ戦略に従った軌跡をそれぞれ表わす．図 5.9(a) と図 5.9(b) から，全ての推定結果を用いた場合はエージェント 0 の軌跡はほぼ一致する．しかし，エージェント 0 の近傍に位置する下側の軌跡と戦略は，評価データと異なる．次に図 5.9(a) と図 5.9(c) から，エージェント 0 の近傍に位置する下側の軌跡と戦略は評価データと一致する．しかし，エージェント 0 の軌跡は評価データと異なる．そのため，エージェント 0 についての推定結果は正確とはいえない．

最後に，表 5.6 に，エージェント 17 に対して推定された戦略を示す．表には，図 5.5(a)，図 5.5(b) に示す二種類の初期位置ごとに推定に成功した結果を示している．この結果から，エージェント 1 体の軌跡に対して，単純な戦略から複雑な戦略まで複数の異なる戦略が推定されていることがわかる．また，初期位置ごとに推定される戦略も異なり，評価データ [d,f],[w,η] では共通する推定結果が存在しないことがわかる．

推定結果の可読性

図 5.10 に，提案法に AIC を用いた場合および用いない場合について，それぞれの解から式 (5.7) で求めたノード数の分布を示す．結果から，評価データ [d],[w] では推定された戦略のノード数に差は見られない．一方，その他の評価データでは，推定された戦略のノード数は AIC を用いた方が同等もしくは少ないことがわかる．

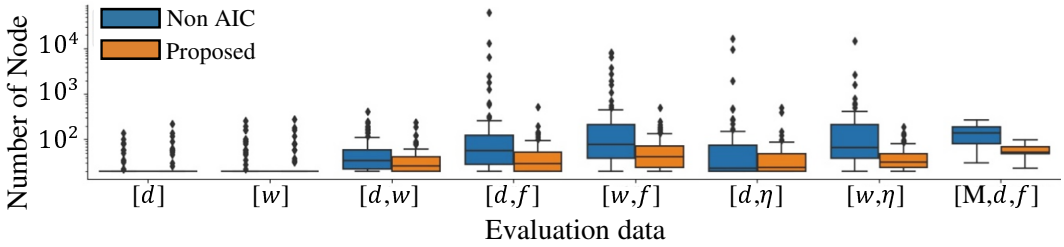


図 5.10: ノード数に対する AIC の影響

5.4 考察

5.4.1 既存法による推定の限界

既存法は，同種の戦略で生成した評価データだけでなく，異種の戦略で生成した評価データ [d,w],[d,η] でも一部推定に成功した．この理由として，エージェントの初期位置の影響が考えられる．評価データ [d,w] の場合，図 5.5(a) のとき 10 試行すべてで推定に成功し，図 5.5(b) のとき 10 試行すべてで推定に失敗した．図 5.5(a) の場合，全エージェントは幅 w が最大の Exit1 付近に位置している．そのため，最短距離 d，幅最大 w のどちらの戦略でも Exit1 を選択するため，同種の戦略で軌跡が再現できるといえる．一方，図 5.5(b) の場合，部屋全体にエージェントが散らばっている．そのため，最短距離の出口が幅最大の出口と一致しないエージェントが存在し，同種の戦略で軌跡を再現できないといえる．

一方，評価データ [d,η] の場合，図 5.5(b) のとき 10 試行すべてで推定に成功し，図 5.5(a) のとき 10 試行すべてで推定に失敗した．図 5.5(b) の場合，混雑を避けるエージェント $15, 16, 17, 18, 19$ は，各エージェントから最短距離に位置する出口が最も混雑度が低い．そのため，戦略が最短距離 d，混雑度最小 $d+\eta$ のいずれでも最短距離の出口を選択するため，同種の戦略で軌跡が再現できるといえる．一方，図 5.5(a) の場合，混雑を避けるエージェント $15, 16, 17, 18, 19$ が群衆の外側に位置している．そのため，最短距離の出口が混雑度最小の出口と一致しないエージェントが存在し，同種の戦略で軌跡を再現できないといえる．

したがって，異種の戦略で振舞う軌跡に対して，その軌跡を同種の戦略を用いて再現可能な場合を除き，既存法による推定は困難である．

5.4.2 提案法による推定の限界

表 5.3，表 5.4 から，提案法は同種または異種のどちらの戦略で振舞う軌跡に対して推定可能といえる．また，図 5.10 からノード数が少なく，可読性を考慮した戦略が推定可能であるといえる．しかし，表 5.5 および表 5.6 の結果は，エージェントの戦略推定が不良設定問題，すなわち，ある軌跡を再現する戦略が複数存在することを示唆している．この結果は，既存法 [Zhong 14] においても確認されている．

そのため，人がただ一つの真の戦略を持ち，その戦略を正しく推定するには，単純に推定法を適用するだけでは難しい．例えば，複数の軌跡を用いる方法が考えられる．人が真の戦略を持つならば，部屋および初期位置といった環境が異なる場合でも，推定される戦略は変化しないはずである．したがって，同じエージェント群を複数の異なる環境で動かし，それぞれの軌跡から推定した戦略候補の積集合で真の戦略を推定する方法が考えられる．しかし，既存法および提案法はともに，軌跡を再現する戦略候補の網羅的な推定を保証していない．そのため，戦略候補の積集合の存在，およびその集合に真の戦略が含まれる保証もできない．したがって，既存法，提案法ともに，軌跡に基づく真の戦略の推定は困難である．

5.4.3 実データ適用に向けた課題

表 5.3，表 5.4 から，手動操作の軌跡を含む評価データ [M,d,f] に対して，既存法，提案法ともに推定に失敗した．そのため，手動操作を含む実データに適用時，推定精度をあげるためには幾つかの改善が必要となる．

推定精度向上のための方法は二つ考えられる．一つめは，推定誤差の評価方法の緩和である．例えば，図 5.8 に示した出口の通過誤差 D_p を用いて緩和した場合，その誤差は非常に小さいことから，推定精度は向上する．また，ランダム性を考慮して推定誤差を定義することで，推定精度が向上する可能性もある．

二つめは，人の観測する特徴量の特定である．図 5.4 に示したように，手動操作の軌跡は，初期位置では出口 Exit2 を選択し，移動の途中で最短距離の出口 Exit3 へ選択を変更している．すなわち，この軌跡を正確に再現するには，この選択が変わった要因を特定する必要がある．例えば，特徴量が視界に依存する場合，エージェントは最短距離の戦略をもつものの，Exit3 に関する特徴量の観測が遅れた，と解釈できる．ただし，特殊な特徴量が増えた場合，その組み合わせである戦略の可読性が悪化する可能性には考慮が必要である．

したがって，手動操作を含む実データに提案法を適用する場合，推定誤差の緩和，または人の観測する特徴量について十分な検討が必要である．

5.5 結言

本章では，群衆シミュレーションに不可欠な人の行動則のうち，目標状態の選択基準である「戦略」に着目し，エージェントの軌跡に基づき戦略を推定する方法を提案した．本手法は，既存法 [Zhong 14] が「全ての人が同種の戦略を持つ」ことを前提にしているのに対して,「人によって戦略が異なる」ことを前提とした異種の戦略推定法である．

計算機実験では，既存法では推定が困難な軌跡を示し，その場合でも提案法により推定可能なことを確かめた．また，戦略が異なっていても，観測される軌跡が一致する場合があることを示し，多様な戦略を獲得できる一方で，人の持つ真の戦略を特定することへの限界を

示した．加えて，実際の軌跡へ適用する場合の課題として，推定誤差の緩和，もしくは人の観測する特徴量の設計が重要であることを示した．

提案法の拡張可能性として，エージェントごとの戦略が切り替わる場合，戦略を一般化して行動則を推定する場合が考えられる．[浪越 17] では，戦略が切り替わる場合へ提案法を拡張している．この拡張法では，戦略の選択を追加した三段階の意思決定において，環境の特徴量から各戦略を評価する関数を戦略と同時に推定することで，戦略の切り替えを実現している．また，提案法は，木構造と木構造に従うエージェントのグループを推定するため，エージェントの観測から行動を出力する関数を木構造として表現できれば，提案法の適用が可能である．

一方，エージェントの意思決定が確率的な場合には直接拡張できない．なぜなら，式 (5.1) で表される評価関数は，戦略に従うシミュレーション結果が一意に定まることを仮定しているからである．そのため，確率的な振舞いを扱う場合は，(1) 確率分布を木構造で表現し，(2) シミュレーションでエージェントの軌跡を複数回サンプルし，そのサンプルを評価する適応度関数を定義することで提案法を拡張するか，完全観測環境を対象に次章からのマルチエージェント逆強化学習を適用する必要がある．

次章からは，ABM における振舞い予測とは異なり，マルチエージェント強化学習において望ましい振舞いを獲得する報酬関数の設計法について述べる．本章と次章では，エージェントの軌跡を用いた推定法である点が共通するが，エージェントのもつ行動則ではなく，エージェントに与えるインセンティブを推定する点で異なる．

第6章　並列座標降下法を用いた報酬の学習速度改善

本章では，単一エージェントの強化学習に対する報酬設計問題のアプローチとして用いられている逆強化学習を，マルチエージェント系に導入する方法を提案する．提案法は，マルコフ決定過程を前提とした逆強化学習を，マルコフゲームに拡張したマルチエージェント逆強化学習（MAIRL : Multi-agent Inverse Reinforcement Learning）として位置付けられる．

一般に，逆強化学習では，エージェントの「状態と行動の組合せ」の系列（軌跡）の集合を入力として,「報酬の推定」と，推定された報酬を用いた「最適方策の探索」の二つの過程を，方策が生成する軌跡と元の軌跡が一致するまで繰り返す．後者の「最適方策の探索」については，強化学習が用いられ，Inner Loop の強化学習と呼ばれる．

マルチエージェント系への逆強化学習の導入において以下の三つの主な課題が考えられる．

(1) 「全エージェントの報酬推定」と「Inner Loop の強化学習による最適方策の探索」それぞれに対する計算量が膨大であること

(2) MARL 特有の課題 [Hernandez-Leal 19] の一つである同時学習の問題を，MARIL はそのまま継承すること

(3) エージェント数に対し Inner Loop の探索空間が指数関数的に増大すること

既存の MAIRL のうち，上記の（1）に対しては，モデルフリー MAIRL[Wei 19, Yu 19] がある．この手法では，双対上昇法で方策と報酬を交互に更新していくため，報酬更新ごとの Inner Loop の MARL は不要であるが，アルゴリズム全体では MARL を一度実行しており，前述した（2）や（3）は避けられない．

（2）に対しては，座標降下法によるアプローチ [Ziebart 10b, Yang 20b] がある．この手法では，報酬の更新をエージェント 1 体ごとに限定して行う．すなわち，各エージェントの最適方策を，他エージェント方策を固定して計算し，単一エージェントの強化学習問題に帰着することによって，同時学習問題を回避している．しかし，報酬推定と方策の探索をエージェントごとに巡回更新するため,（1）や（3）の計算量の問題は残される．

そこで本章は，後者の座標降下法に着目し,（3）を軽減し学習速度を改善するため，MAIRL に並列座標降下法を導入した方法を提案し，学習速度を改善できることを実験的に示す．提案法は，座標降下法における同期的並列化 [Wright 15] と同様に，全エージェントの方策を定期的に同期しながら，各エージェントの報酬と方策の更新については複数のプロセッサによる並列処理が可能な方法であり，並列化による高速化の実現を想定している．

以下，6.1 節で座標降下法に基づくマルチエージェント逆強化学習について述べ，6.2 節で提案法を示したのち，6.3 節で計算機実験，6.4 節で考察，6.5 節で本章をまとめる．

6.1 座標降下法に基づくマルチエージェント逆強化学習

マルチエージェント逆強化学習は，報酬を未知とするマルコフゲームとエキスパート軌跡 $\tau = (s_t, \boldsymbol{a}_t)_{t=0}^{T}$ の集合 $\mathcal{D}^E = \{\tau_m\}_{m=1}^{M}$ から全エージェントの報酬 $\boldsymbol{r}$ を推定する問題である．以下では，2.5 節で述べた Maximum Discounted Cuasl Entorpy IRL（MDCE IRL）をマルコフゲームへ拡張した定式化を示した後，座標降下法を用いた MAIRL の関連研究を整理する．

6.1.1 定式化

MDCE IRL をマルコフゲームに拡張した Multi-agent MDCE IRL を式 (6.1) から式 (6.4) に示す．この定式化は，各エージェント i について，他エージェントがエキスパート方策に従う場合の特徴ベクトル $\boldsymbol{f}_i : \mathcal{S} \times \mathcal{A} \to \mathbb{R}^k$ の割引期待値 $\overline{\boldsymbol{f}}_{i,\pi_i,\pi^E_{-i}}$ がエキスパート方策の割引特徴期待ベクトル $\overline{\boldsymbol{f}}_{i,\boldsymbol{\pi}^E}$ と一致する方策 $\boldsymbol{\pi}$ の獲得問題を表す．このとき，状態遷移確率 P を直接知ることはできないが，シミュレータ環境における試行錯誤は可能とする．

$$\max_{\boldsymbol{\pi}} \quad \sum_{i=1}^{N} \alpha H_{\pi_i,\pi^E_{-i}}(\pi_i) \tag{6.1}$$

$$\text{s.t.} \quad \overline{\boldsymbol{f}}_{i,\boldsymbol{\pi}^E} = \overline{\boldsymbol{f}}_{i,\pi_i,\pi^E_{-i}} \quad \forall i \in [1, N] \tag{6.2}$$

$$\pi_i(a_i|s) \geq 0 \quad \forall a_i \in \mathcal{A}_i, s \in \mathcal{S}, i \in [1, N] \tag{6.3}$$

$$\sum_{a_i \in \mathcal{A}_i} \pi_i(a_i|s) = 1 \quad \forall s \in \mathcal{S}, i \in [1, N] \tag{6.4}$$

ここで，式 (6.1) はエージェントごとの方策に対するエントロピーの総和であり，そのエントロピーは式 (6.5) で定義される．式 (6.2) は式 (6.6) の割引特徴期待ベクトル $\overline{\boldsymbol{f}}_i$ を一致させる制約，式 (6.3)，式 (6.4) は方策に関する制約を表す．

$$H_{\boldsymbol{\pi}}(\pi_i) \triangleq \mathbb{E}_{P_0,P,\boldsymbol{\pi}}\left[\sum_{t=0}^{\infty} -\gamma^t \log \pi_i\left(a_{t,i}|s_t\right)\right] \tag{6.5}$$

$$\overline{\boldsymbol{f}}_{i,\boldsymbol{\pi}} \triangleq \mathbb{E}_{P_0,P,\boldsymbol{\pi}}\left[\sum_{t=0}^{\infty} \gamma^t \boldsymbol{f}_i\left(s_t, \boldsymbol{a}_t\right)\right] \tag{6.6}$$

式 (6.1) における方策のエントロピーと式 (6.2) における割引特徴期待ベクトルは，エージェント間で互いに独立なため，エージェント i のラグランジュ緩和問題は，式 (2.11) と同様に式 (6.7) で表される．

$$\min_{\boldsymbol{\theta}_i}\left[\max_{\pi_i} \alpha H_{\pi_i,\pi^E_{-i}}(\pi_i) + \boldsymbol{\theta}_i^{\top}\overline{\boldsymbol{f}}_{i,\pi_i,\pi^E_{-i}}\right] - \boldsymbol{\theta}_i^{\top}\overline{\boldsymbol{f}}_{i,\boldsymbol{\pi}^E} \tag{6.7}$$

Inner Loop は，元の状態遷移確率 P を他エージェントのエキスパート方策 π^E_{-i} で周辺化した状態遷移確率で定義される単一エージェント環境であり，エキスパート方策 $\boldsymbol{\pi}^E$ が既知の場合は，各エージェントごとに MDCE IRL と同様の手順で解くことができる．

6.1.2 既存法

7.2.1 節の式 (6.7) は，エキスパート方策 $\boldsymbol{\pi}^E$ を必要とすることを示しているが，状態・行動空間の大きい場合などでは，エキスパート方策を十分近似できる軌跡集合を得ることは容易ではない．そこで，座標降下法を用いた MAIRL は，エージェントごとに報酬と方策を更新し，他エージェント方策にそれまでに Inner Loop で得た学習中の方策 $\boldsymbol{\pi}^{\mathrm{learn}}$ を用いる．

Ziebart ら [Ziebart 10b] は状態遷移確率が既知，かつ Finite-horizon のマルコフゲームを前提として，Inner Loop には動的計画法，報酬更新には勾配降下法を用いて，エージェントごとの式 (6.7) を更新している．Yang ら [Yang 20b] は，Linear IRL[Ng 00] をマルコフゲームに拡張した定式化 [Reddy 12] において，式 (6.8)，式 (6.9) で表されるエージェントごとの問題を巡回更新する．

$$\max_{\boldsymbol{\theta}_i} \quad V_{i,\boldsymbol{\pi}^E}(s_0) - V_{i,\pi_i,\pi^E_{-i}}(s_0) \tag{6.8}$$

$$\text{s.t.} \quad \|\boldsymbol{\theta}_i\|_2 \leq 1 \tag{6.9}$$

ここで s_0 は初期状態，V_i はエージェント i の状態価値関数を表し，任意の方策 π_i とエキスパート方策 $\boldsymbol{\pi}^E$ の状態価値の差が最も大きくなる報酬を求めている．ただし，この定式化では方策のエントロピー項を考慮していない（$\alpha = 0$）．

いずれの手法も，エージェントの報酬と方策を巡回更新する座標降下法を用いている．これに対して，本提案は，学習速度の改善を目的とした，状態遷移確率未知の環境で式 (6.7) の並列化を可能にする方法を考える．

6.2 提案法：並列座標降下法による解法

本節では MAIRL に座標降下法を導入する．以下では，提案法と同じく最大エントロピー原理に従う Ziebart ら [Ziebart 10b] の手法を既存法と位置づけ，座標降下（Coordinate Descent）法を用いる既存法を CD 法，並列座標降下（Parallel CD）法を用いる提案法を PCD 法と記す．以下では，まず CD 法と PCD 法を更新手順の点から比較し，並列化の導入可能性についてまとめたのち，提案法のアルゴリズムを示す．

6.2.1 更新手順概要

図 6.1(a)，(b) は，CD 法と PCD 法の更新手順を比較するための概念図である．CD 法は，エージェントを 1 体ずつ選択して報酬を δ 回更新し，その最適方策を他エージェント方策と

して固定する．一方，PCD 法は，複数のエージェントの報酬を δ 回ずつ並列に更新し，各最適方策を疑似方策 $\boldsymbol{\pi}^{\text{pseudo}}$ に書き換えた上で同期的に共有する．

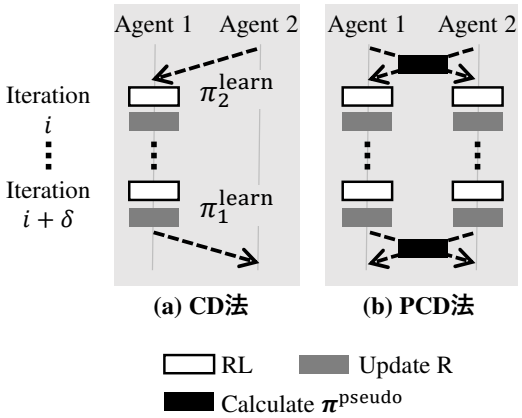


図 6.1: エージェント 2 体の場合の更新手順

6.2.2 並列化の導入可能性

エージェント i の疑似方策 π_i^{pseudo} は式 (6.10) で定義される．

$$\pi_i^{\text{pseudo}} = \begin{cases} \frac{\#(s,a_i)}{\#(s)} & s \in \mathcal{D}^E \\ \pi_i^{\text{learn}} & \text{otherwise} \end{cases} \tag{6.10}$$

ここで，$\#(s,a_i)$ と $\#(s)$ は，それぞれ，状態行動の組 (s,a_i) と状態 s がエキスパート軌跡集合 $\mathcal{D}^E$ に含まれる数を表す．疑似方策は，それまでに学習したエージェント i の方策 π_i^{learn} を，エキスパート軌跡の行動分布 $\frac{\#(s,a_i)}{\#(s)}$ で上書きした方策であり，ここで，他エージェントは，それぞれに対応するエキスパートの方策を実行しているものとする．

十分な数のエキスパート軌跡が用意できれば，疑似方策は，式 (6.7) で得られるエキスパート方策に一致することから，各エージェントは，この疑似方策を，エキスパート方策に代替することによって，他エージェントへの問い合わせをすることなく，ローカルに報酬と方策の更新を繰り返すことができる．すなわち，各エージェントは疑似方策を用いて，独立に更新できるため，並列化が可能になる．

6.2.3 アルゴリズム

Algorithm 3 に提案法のアルゴリズムを示す．3 行目では，全てのエージェントについての疑似方策 π_i^{pseudo} を式 (6.10) で求める．4,5,6 行目は各エージェントごとに並列で実行され，4 行目で Algorithm 4 の Soft Q-Learning[Zhou 18] で方策 π_i^{learn} を学習したのち，5 行目でその方策から軌跡を生成，6 行目でその軌跡とエキスパート軌跡の割引特徴期待ベクトルの差 $\nabla\boldsymbol{\theta}_i$ から勾配降下法により報酬の重み $\boldsymbol{\theta}_i$ を更新する．ただし重み更新は，割引特徴期待

値の絶対値差の最大値 $\max_k \left|\overline{\boldsymbol{f}}_{k,i,\boldsymbol{\pi}^E} - \overline{\boldsymbol{f}}_{k,i,\pi_i,\pi_{-i}^{\text{pseudo}}}\right|$ が十分小さくなった場合か，打ち切り回数に達した場合に更新を打ち切る．全体の収束条件は，割引特徴期待ベクトルについて最も絶対値差の大きい次元の値 $\max_k \left|\overline{\boldsymbol{f}}_{k,i,\boldsymbol{\pi}^E} - \overline{\boldsymbol{f}}_{k,i,\boldsymbol{\pi}_i^{\text{learn}}}\right|$ が閾値 ϵ 以下になった場合とする．

Algorithm 3 MAIRL with parallel coordinate descent

Input: Markov game$\backslash \boldsymbol{r}$, Expert trajectories $\mathcal{D}^E$
Output: Reward weight $\boldsymbol{\theta}$

1: Initialize policies π_i and $\theta_i \quad \forall i \in [1, N]$
2: **for** Iteration $j = 1, 2 \ldots$ **do**
3: Calculate $\pi_i^{\text{pseudo}} \quad \forall i \in [1, N]$
4: **for** $i \in [1, N]$ **do** (in parallel)
5: Learn $\pi_{i,j}$ with Algorithm 4 on fixed π_{-i}^{pseudo}
6: Sample $\mathcal{D}_i$ from $\pi_{i,j}, \pi_{-i}^{\text{pseudo}}$
7: $\nabla \boldsymbol{\theta}_i \leftarrow \mathbb{E}_{\mathcal{D}_i}\left[\sum_t \gamma^t f_i(s_t, \boldsymbol{a}_t)\right] - \overline{f}_{i,\boldsymbol{\pi}^E}$
8: **end for**
9: **end for**

Algorithm 4 Soft Q-learning

Input: Reward weight $\boldsymbol{\theta}_i$, Explore policy π_i^{rnd}, Other agent's policy π_{-i},
Entropy coefficient α

1: **for** episode $= 1, 2, \cdots$ **do**
2: **while** reach terminal state or max step **do**
3: Generate sample $(s_t, \boldsymbol{a}_t, s_{t+1})$ from $\pi_i^{\text{rnd}}, \pi_{-i}$
4: $V_i(s_{t+1}) \leftarrow \alpha \log \sum_{a_i \in \mathcal{A}_i} \exp\left(\frac{1}{\alpha} Q_i\left(s_{t+1}, a_i\right)\right)$
5: $Q_i\left(s_t, a_{t,i}\right) \leftarrow Q_i\left(s_t, a_{t,i}\right) + \quad \eta_t\left[\boldsymbol{\theta}_i^\top \boldsymbol{f}_i\left(s_t, \boldsymbol{a}_t\right) + \gamma V_i(s_{t+1}) - Q_i\left(s_t, a_{t,i}\right)\right]$
6: **end while**
7: **end for**

6.2.4 計算量評価

エージェント数 N に対する提案法の実行計算量を評価する．最も計算量の多い Inner Loop は，Algorithm 3 の 5 行目に該当する．そのため，1 [iteration] あたりの Inner Loop の実行回数は CD 法の場合は $O(N)$，PCD 法の場合，十分な計算資源のもとで $O(1)$ となる．一方，Inner Loop の計算コストは，エージェント数に対して指数オーダーで増大する状態行動空間において，全ての状態・行動の組を十分に訪問する必要があることから，対象問題によっては実行計算量に与える影響が大きい．

(a) GN1：決定的遷移　　(b) GN2：確率的遷移

図 6.2: 環境とエキスパート軌跡（エージェント 1, 2 に対して $x_1^{\mathrm{S}}, x_2^{\mathrm{S}}$ は初期座標，$x_1^{\mathrm{G}}, x_2^{\mathrm{G}}$ はゴール座標．GN2 の太線は障壁を表す.）

6.3 計算機実験

提案法の性能を評価するための計算機実験では，対象とする環境を離散の状態と行動空間とし，エージェントは環境に対して完全観測，報酬については General-sum タイプのベンチマーク問題である Grid Navitaion を取り上げる．本対象において，エキスパートは，特定の均衡解を実行するとしたとき，単純な座標降下法に基づく CD 法と比べ，提案法する PCD 法により学習速度が改善できることを示す.

6.3.1 実験設定

図 6.2 に示す Grid Navigation は，3×3 のグリッド上で 2 体のエージェントが衝突を回避しながら各自のゴール座標を目指す [Hu 03, Bowling 03].

実験には，ゴール座標と状態遷移確率が異なる 2 種類の環境を用いる．以下では，図 6.2(a), 図 6.2(b) の環境をそれぞれ GN1，GN2 と記す．いずれの環境も，状態は全エージェントの座標の組合せ，行動は隣接するグリッドへの移動 $\mathcal{A}_1 = \mathcal{A}_2 = \{\mathrm{up}, \mathrm{down}, \mathrm{right}, \mathrm{left}\}$，初期座標は図中の $x_1^{\mathrm{S}}, x_2^{\mathrm{S}}$ とする．

状態遷移確率は GN1 は決定的，GN2 は確率的な遷移環境とし，その違いは初期座標のグリッドと一つ上のグリッドの間の障壁により生じる．障壁のある GN2 は，x_1^{S} もしくは x_2^{S} で up が選択された場合に 50% の確率で遷移に失敗する．その他は全て決定的に遷移し，グリッド外に向けて移動する場合と両方のエージェントが同じグリッドへ移動する場合[1]だけ遷移前の座標に留まる．いずれかのエージェントがゴール座標 x_i^{G} へ到達した状態からは，任意の行動の組合せで吸収状態へ遷移しエピソードを打ち切る．

エキスパート軌跡は，決定的な Nash 均衡方策から生成する．エキスパートが従う真の報酬は，ゴール座標に到達したエージェントに $+100$，両エージェントが衝突した場合に -1 を与えた場合を考える．この真の報酬において，図 6.2 の矢印に沿う軌跡は Nash 均衡解の一つである [Hu 03]．そこで，吸収状態に最短で遷移できるステップ数を最大ステップ数として，各矢印に沿って 10^3 本のエキスパート軌跡を生成した.

[1]いずれかのエージェントがゴール座標へ移動する場合を除く

その他，報酬の特徴ベクトルは状態 s と行動の組合せ $\boldsymbol{a}$ に対する one-hot ベクトル $\boldsymbol{f}(s, \boldsymbol{a})$，割引率は 0.9 とし，割引特徴期待ベクトルは 10^3 本の軌跡で近似する．重み $\boldsymbol{\theta}_i$ の初期値は各次元ごとに 0 から 1 の一様分布でサンプリングし，初期方策は一様分布とする．報酬の勾配には 0.01 で重み付けした L2 正則化を加え，ステップ幅 0.1 の最急降下法で更新する．Inner Loop の Soft Q-Learning は，最短で終端状態に辿り着くステップ数を打ち切りステップとして 10^3 エピソード学習する．エントロピー項の係数 α は決定的方策を模倣することから 0.01 とした．収束条件の閾値 ϵ は，状態遷移確率のランダム性を考慮して，GN1 では 10^{-10}，GN2 では $4 \cdot 10^{-2}$ とした．

6.3.2 実験結果

CD 法，PCD 法のそれぞれについて，他エージェント方策にランダム方策 π^{rnd}，学習中の方策 π^{learn}，疑似方策 π^{pseudo} を用いた場合を比較する．

図 6.3(a)，図 6.3(b)，図 6.3(c) に，それぞれ他エージェント方策ごとの推定誤差の推移を表す．また，表 6.1 には，収束時の推定誤差と獲得報酬を示す．推定誤差は，各方策の割引特徴期待ベクトルのユークリッド距離であり，6.3.1 節で述べたように，特徴ベクトルが状態 s と行動の組合せ $\boldsymbol{a}$ の one-hot ベクトルなため，エキスパート方策と推定方策の $(s, \boldsymbol{a})$ の頻度の差を意味する．獲得報酬は，推定方策による真の報酬の割引累積報酬であり，ゴール座標への到達量とエージェントの衝突量を表す．いずれも，推定方策からサンプリングした 10^3 本の軌跡を用いて計算している．

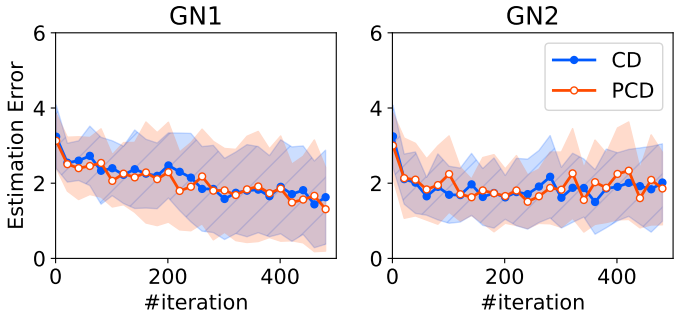
他エージェント方策がランダム方策 π^{rnd} の場合は，いずれの手法も GN1 で約 70% の試行で収束せず，GN2 ではエキスパートに一致する解が得られなかった．獲得報酬で比較した場合，どちらのエージェントもエキスパート方策の獲得量の半数以下であり，ゴール座標へ到達できていない．この結果は，他エージェント方策が収束先に大きく影響することを示している．

次に，他エージェント方策が学習中の方策 π^{learn} の場合は，GN1 でいずれの手法もエキスパートと同等の結果に収束し，GN2 で π^{rnd} の場合よりエキスパートに近い結果が得られた．一方，CD 法と PCD 法について学習速度を比較したとき，図 6.3(b) から GN2 では差が見られるが，表 6.1 から GN1 ではほぼ同等であった．この結果は，PCD 法の報酬の更新回数が CD 法と比べて 2 倍多いのにも関わらず，PCD 法に効果がない場合があることを示している．

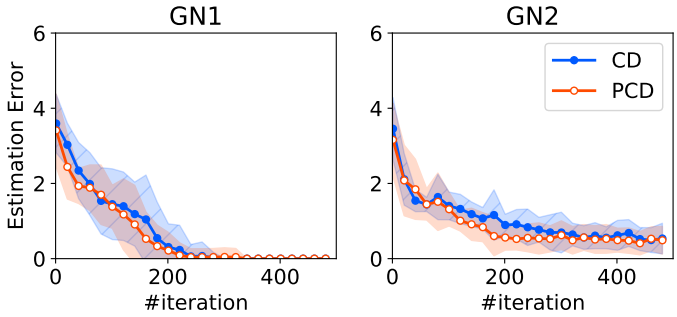
一方，他エージェントの方策が疑似方策 π^{pseudo} の場合は，どちらの手法も学習速度が改善したうえに，GN1 では PCD 法が CD 法の約半分の繰返し数で収束した．また獲得報酬で比較すると，表 6.1 から，いずれの手法もエキスパートとほぼ一致する解が得られている．

最後に，獲得した報酬の重みを比較する．図 6.4 に学習中の方策を用いた CD 法と，疑似方策を用いた PCD 法で推定した重みの分布の比較を示す．重みの初期値が 0 から 1 であるのに対し，推定された重みが 1 以上の状態・行動の組合せはいずれもエキスパート軌跡に含

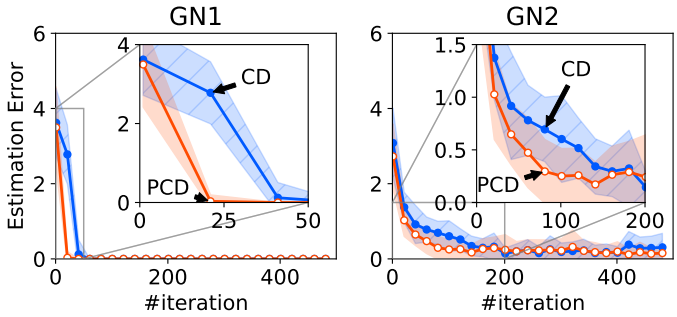
まれており，この傾向は CD 法，PCD 法で差がなかった．一方，疑似方策を用いない CD 法は，疑似方策を用いた PCD 法に比べて，負の重みを大きく推定する傾向がみられる．



(a) ランダム方策 π^{rnd} を用いた場合



(b) 学習中の方策 π^{learn} を用いた場合



(c) 疑似方策 π^{pseudo} を用いた場合

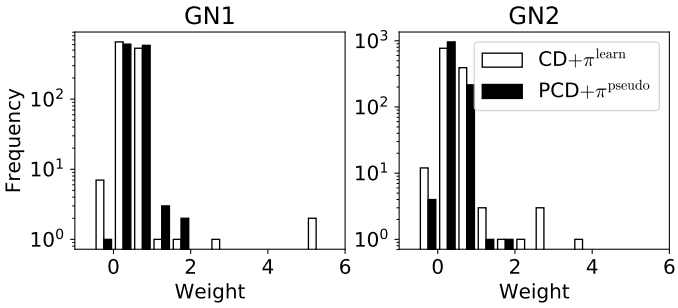
図 6.3: 推定誤差の推移（20 [iteration] ごとの 30 試行平均と標準偏差を表す.）

6.4 考察

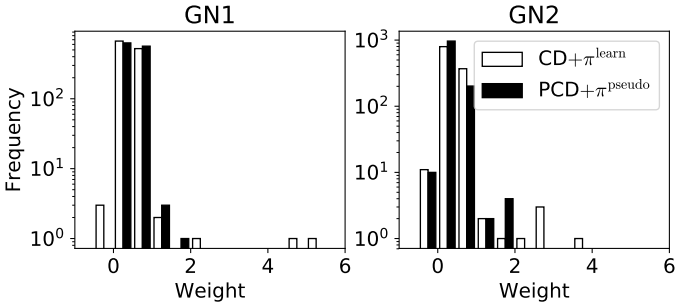
6.4.1 並列化が有効に働かない場合とその理由

他エージェント方策に学習中の方策を用いた場合，特に GN1 で CD 法と PCD 法の学習速度に差が見られなかった．この原因は PCD 法の更新手順にある．

図 6.1 における方策の更新手順に着目すると，PCD 法は二つの独立した最適反応列を含んでいる．具体的には，エージェント i の方策 π_i に対するエージェント j の最適反応を $\sigma_j(\pi_i)$ すると，それぞれ $(\pi_1, \sigma_2(\pi_1), \sigma_1(\sigma_2(\pi_1)), \ldots)$，$(\pi_2, \sigma_1(\pi_2), \sigma_2(\sigma_1(\pi_2)), \ldots)$ と表せる．しか



(a) Agent1 の重み $\boldsymbol{\theta}_1$



(b) Agent2 の重み $\boldsymbol{\theta}_2$

図 6.4: 推定報酬の重み分布の例

表 6.1: 収束時の推定誤差とエージェントごとの獲得報酬（Rate は収束した試行の割合，#iteration は収束しなかった場合を除いた収束時の繰返し数，Error は打ち切り時の推定誤差，R_1, R_2 はエージェント 1, 2 ごとの獲得報酬をそれぞれ表す．全て 30 試行の平均と標準偏差であり，GN2 は最後の 10 [iteration] のうち，推定誤差が最小のときの値を用いた.）

	GN1				GN2		
	Rate [%]	#iteration	R_1	R_2	Error	R_1	R_2
CD+π^{rnd}	30	304.2 ± 102.9	28 ± 36	26 ± 36	0.84 ± 0.15	17 ± 22	55 ± 25
PCD+π^{rnd}	33	260.5 ± 110.7	33 ± 37	24 ± 35	0.84 ± 0.17	27 ± 24	47 ± 26
CD+π^{learn}	**100**	159.9 ± 55.3	$\mathbf{73} \pm 0$	$\mathbf{73} \pm 0$	0.23 ± 0.23	36 ± 13	78 ± 6
PCD+π^{learn}	**100**	157.1 ± 52.0	$\mathbf{73} \pm 0$	$\mathbf{73} \pm 0$	0.23 ± 0.21	42 ± 4	75 ± 15
CD+π^{pseudo}	**100**	35.0 ± 4.9	$\mathbf{73} \pm 0$	$\mathbf{73} \pm 0$	$\mathbf{0.04} \pm 0.01$	$\mathbf{39} \pm 1$	$\mathbf{81} \pm 0$
PCD+π^{pseudo}	**100**	$\mathbf{18.3} \pm 2.5$	$\mathbf{73} \pm 0$	$\mathbf{73} \pm 0$	$\mathbf{0.04} \pm 0.01$	$\mathbf{39} \pm 1$	$\mathbf{81} \pm 0$
Expert	-	-	73 ± 0	73 ± 0	0.03 ± 0.01	38 ± 40	81 ± 0

し，最適反応列ごとの収束までにかかる繰り返し数は，CD 法の結果から分散が大きい．そのため，すべての最適反応列が収束するまで全体としても収束しないことから，単純に並列化するだけでは効果がない場合がある．

6.4.2　疑似方策を用いた並列化の有効性

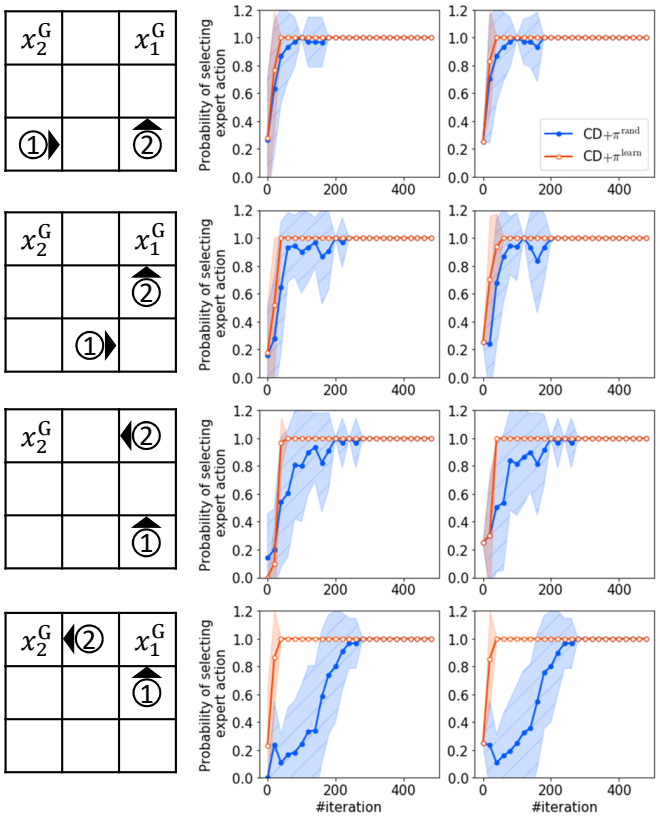


図 6.5: 推定方策がエキスパートと同じ行動をとる確率の推移（左列は状態と各エージェントのエキスパート行動，中央列は Agent1，右列は Agent2 の 30 試行の平均と分散の推移を表す.）

実験結果から，疑似方策を用いた場合，学習中の方策を用いた場合に比べて学習速度を改善し，並列化も有効に働くことがわかった．しかし，他エージェントの方策がもたらす具体的な影響は，実験結果だけからは読み取れない．本節ではその詳細を考察する．

他エージェント方策が及ぼす影響は，報酬の更新か，方策の更新のいずれかである．前者は，疑似方策の有無にかかわらず，エキスパート軌跡上の報酬を増加させる傾向がみられた点で，適切に更新されている．最大エントロピー原理に基づく逆強化学習は，式 (6.7) に示したように，エキスパートを模倣するまで軌跡上の報酬の重みを増やし，エキスパートから逸脱した場合は重みを減らすことから，この結果は妥当といえる．よって，学習速度は，更新されたエキスパート軌跡上の報酬を獲得できるかに左右される可能性が高い．

一方，学習中の方策を比較したとき，疑似方策と学習中の方策ではエキスパートとの一致率の推移が異なっていた．図 6.5 に，GN1 におけるエキスパート方策と推定方策の一致率を，学習中の方策を用いた CD 法と，疑似方策を用いた CD 法で比較した結果を示す．この結果から，疑似方策を用いない場合，ゴールに近い状態ほどエキスパートと一致するのに数多くの繰り返しを必要としている．この方策を他エージェント方策として用いると，エキスパート軌跡上の報酬を獲得することができない．そのため，エキスパート方策と一致する方策を得るには，最悪の場合，エキスパート軌跡に含まれない状態・行動の組合せの全てに十分小さな報酬を推定し，エキスパートからの逸脱を防ぐ必要がある．一方で，疑似方策はエキスパートとほぼ同一の行動分布をとるため，エキスパート軌跡上に十分な報酬が推定されればエキスパート方策を獲得できる．

そのため，MAIRL における学習速度は主に方策の更新に依存しており，疑似方策は他エージェントがエキスパート行動から逸脱するのを防ぐことで，並列化が有効に働くだけでなく，学習速度自体を大きく改善することができる．

6.5 結言

本章では，エージェントごとに報酬と方策を更新する MAIRL において，その並列化法を提案した．計算機実験では，並列化による学習速度の向上を確かめるため，最も単純なベンチマークであるエージェント 2 体の Grid Navigation を用いて，特定の均衡解を再現する方策の獲得と報酬を推定するタスクを用いた．

実験の結果によれば，単純に並列化しても学習速度が改善しない場合があり，この原因は，各エージェントのローカルな処理で必要となる他エージェントの方策がエキスパート方策と異なり，その差異がボトルネックになっていることを確認した．このボトルネックに対しては，他エージェントの方策を，疑似方策に代替することによって，同期による待ち時間を解消し，並列化による学習速度改善効果が見込めることも示した．

次章では，報酬と方策が同一のグループ構造を仮定し，座標降下法に基づく MAIRL に EM アルゴリズムを導入することで，グループ構造と報酬を同時に推定する方法を提案する．

第7章　EMアルゴリズムを用いたグループ構造と報酬の同時推定

協調により複雑なタスクを解くマルチエージェント系では，Leader-Follower モデルなど，複雑な振舞いをたかだか数種類のエージェントの行動則で記述できることがある．本章の目的は，前述のような環境を対象に，報酬や方策が同じエージェントをグループとして扱い，状態・行動の軌跡からグループ構造，すなわちエージェントのグループへのメンバーシップと，各グループの報酬をマルチエージェント逆強化学習（MAIRL）により推定することである．既存の MAIRL は，zero-sum 報酬 [Lin 18, Wang 18]，全エージェントに共通の報酬 [Šošić 17]，個々に異なる報酬 [Yu 19, Wei 19] のいずれかを推定時に仮定するため，軌跡に含まれるグループ構造を陽に推定することができない．そこで，本章では，6 章で述べた座標降下法を用いた MAIRL に EM アルゴリズムを導入することで，グループ構造と報酬を同時推定する．

以下，7.1 節で対象問題についてまとめ，7.2 節で提案法，7.3 節で Grid navigation に適用した結果を示し，7.5 節で本章のまとめを述べる．

7.1　マルコフゲームにおけるグループ構造の導入

図 7.1 に軌跡 τ_m の生成分布の概略を図示する．各エージェントは，K 種類のうちいずれかのグループに属し，同じグループに属するエージェントは同一の線形報酬関数の重み $\boldsymbol{\theta}_k$ と方策 π_k に従う．エージェントの属するグループはメンバーシップ関数 $\phi : [1, N] \to [1, K]$ で表し，メンバーシップ ϕ_l で生成される軌跡分布を $p_l(\tau)$ とする．そして，軌跡の生成分布はメンバーシップごとの混合係数 w_l で重みづけした混合分布 $\sum_l w_l p_l(\tau)$ で表す．仮定するグループ数 K はハイパーパラメータでありメンバーシップの数 L は K^N とする．

7.2　提案法：EM アルゴリズムによる解法

グループ構造を仮定した環境で，軌跡を生成したグループ構造とその報酬を推定するため，EM アルゴリズムを導入する．以下では，EM アルゴリズムを用いた定式化を示したのち，提案法のアルゴリズムを示す．

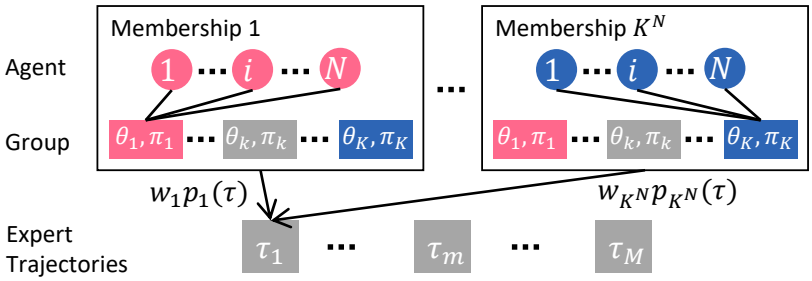


図 7.1: 軌跡の生成分布

7.2.1 定式化

軌跡 τ_m を生成したメンバーシップが観測されていないエキスパート軌跡集合 $\mathcal{D}^{\mathrm{E}}$ からその軌跡を生成した $\{\theta_k\}$ と $\{w_l\}$ を推定する問題を，混合分布に対する EM アルゴリズム [Bishop 08] と同様の手順で定式化する．まず，軌跡 τ_m がメンバーシップ l のもとで生成されたかを表す潜在変数を定義する．

$$z_{ml} = \begin{cases} 1 & (\text{if } \tau_m \text{ sampled from } p_l(\tau)) \\ 0 & (otherwise) \end{cases} \tag{7.1}$$

潜在変数は $\sum_l z_{ml} = 1, z_{ml} \in \{0, 1\}$ を満たし，軌跡 τ_m についての潜在変数ベクトルは $\boldsymbol{z}_m$，全ての軌跡についての潜在変数ベクトルは $\boldsymbol{z}$ と表す．軌跡 τ の生成確率は，メンバーシップごとの混合係数 w_l と報酬の重み θ_k からなるパラメータ Θ で次のように表される．

$$p(\tau_m|\Theta) = \sum_{\boldsymbol{z}_m} p(\tau_m|\boldsymbol{z}_m, \Theta) p(\boldsymbol{z}_m|\Theta) \tag{7.2}$$

$$= \sum_{\boldsymbol{z}_m} \prod_l w_l^{z_{ml}} p_l(\tau_n|\Theta)^{z_{ml}} \tag{7.3}$$

$$= \sum_l w_l p_l(\tau_m|\Theta), \tag{7.4}$$

ベイズの定理から，エキスパート軌跡集合 $\mathcal{D}^{\mathrm{E}}$ のもとでの事後分布は次のように表される．

$$p(\boldsymbol{z}_m|\mathcal{D}^{\mathrm{E}}, \Theta) = \frac{\sum_{\boldsymbol{z}_1} \cdots \sum_{\boldsymbol{z}_{M-1}} \prod_m p(\tau_m, \boldsymbol{z}_m|\Theta)}{\sum_{\boldsymbol{z}_1} \cdots \sum_{\boldsymbol{z}_M} \prod_m p(\tau_m, \boldsymbol{z}_m|\Theta)} \tag{7.5}$$

$$= \frac{p(\tau_m, \boldsymbol{z}_m|\Theta)}{\sum_l' w_l' p_l'(\tau_m|\Theta)} \tag{7.6}$$

$$= \frac{w_l p_l(\tau_m|\Theta)}{\sum_l' w_l' p_l'(\tau_m|\Theta)} \tag{7.7}$$

ただし最後の等式は $z_{ml}=1$ の場合に成り立つ．$\mathcal{D}^{\mathrm{E}}$ とパラメータ $\Theta^{(t)}$ に対する完全データの対数尤度の期待値は次のように表される．

$$
\begin{aligned}
Q(\Theta,\Theta^{(t)}) &= \mathbb{E}_{p(Z|\mathcal{D}^{\mathrm{E}},\Theta^{(t)})}\left[\log p(\mathcal{D}^{\mathrm{E}},Z|\Theta)\right] && (7.8)\\
&= \mathbb{E}_{p(Z|\mathcal{D}^{\mathrm{E}},\Theta^{(t)})}\left[\log \prod_m p(\tau_m,\boldsymbol{z}_m|\Theta)\right] && (7.9)\\
&= \mathbb{E}_{p(Z|\mathcal{D}^{\mathrm{E}},\Theta^{(t)})}\left[\sum_m\sum_l z_{ml}\log w_l p_l(\tau_m|\Theta)\right] && (7.10)\\
&= \sum_m\sum_l \gamma_{ml}^{(t)}\log w_l + \sum_m\sum_l \gamma_{ml}^{(t)}\log p_l(\tau_m|\Theta) && (7.11)
\end{aligned}
$$

ここで $\gamma_{ml}^{(t)}$ は軌跡 τ_m の負担率 $\mathbb{E}_{p(Z|D,\Theta^{(t)})}[z_{ml}]$ を表す．

よって，式 (7.11) から，次の E-step，M-step を交互に更新し全データの対数尤度を改善する．

E-step

$$
\gamma_{ml}^{(t)} \leftarrow \frac{w_l p_l(\tau_m|\Theta^{(t)})}{\sum_{l'} w_{l'} p_{l'}(\tau_m|\Theta^{(t)})} \tag{7.12}
$$

M-step

$$
w_l^{(t+1)} \leftarrow \frac{1}{M}\sum_m \gamma_{ml}^{(t)} \tag{7.13}
$$

$$
\boldsymbol{\theta}_k^{(t+1)} \leftarrow \boldsymbol{\theta}_k^{(t)} + \beta\sum_m\sum_l \gamma_{ml}^{(t)}\nabla_{\boldsymbol{\theta}_k}\log p_l(\tau_m|\{\boldsymbol{\theta}_k\}) \tag{7.14}
$$

次節では，軌跡の尤度関数 $p_l(\tau_m|\Theta^{(t)})$ と報酬の重み $\boldsymbol{\theta}_k$ に関する勾配の計算方法とアルゴリズムを述べる．

7.2.2 アルゴリズム

まず，軌跡の尤度関数を以下のように定義する．

$$
p_l(\tau|\Theta^{(t)}) = \frac{f(\tau,\boldsymbol{\pi})}{Z_l} = \frac{\prod_i \prod_{(s,a_i)\in\tau_{mi}} \pi_{\phi_l(i)}(a_i|s)}{Z_l} \tag{7.15}
$$

Z_l はメンバーシップ l における分配関数を表す．しかし，Z_l は全軌跡集合に関する積分が含まれるため，状態･行動空間が大きいほど計算困難となる．そこで，Guided cost learning[Finn 16] と同様に重点サンプリングを用いて分配関数を近似する．

$$
Z_l = \mathbb{E}_{\tau\sim\mu}\left[\frac{1}{\mu}f(\tau,\boldsymbol{\pi})\right] \approx \frac{1}{|\mathcal{D}_l^{\mathrm{IS}}|}\sum_{\tau\in\mathcal{D}_l^{\mathrm{IS}}}\frac{1}{\mu}f(\tau,\boldsymbol{\pi}) \tag{7.16}
$$

ここで $\mathcal{D}_l^{\mathrm{IS}}$ は μ により生成された軌跡を表し，μ はエキスパート軌跡の生成確率 $\tilde{p}$ と，メンバーシップ l の学習方策で得られたサンプル q の混合分布 $\mu(\tau)=\frac{1}{2}\tilde{p}(\tau)+\frac{1}{2}q(\tau)$ とする．

つぎに，報酬の重み θ_k に関する勾配 $\nabla_{\theta_k} \log p_l(\tau_n|\{\theta_k\})$ を求める．$\phi_l(i) = k$ のとき，式 (2.13) から行動価値関数の勾配は

$$\begin{aligned}\nabla_{\boldsymbol{\theta}_k} V_i(s) &= \nabla_{\boldsymbol{\theta}_k} \log \sum_a \exp Q_i(s,a) \\ &= \sum_a \frac{\exp Q_i(s,a)}{\sum_{a'} \exp Q_i(s,a')} \nabla_{\boldsymbol{\theta}_k} Q_i(s,a) \\ &= \sum_a \pi_i(a|s) \nabla_{\boldsymbol{\theta}_k} Q_i(s,a) \end{aligned} \tag{7.17}$$

で表される．$\Phi_l(k)$ をメンバーシップ l でグループ k に属するエージェント集合とすると，他エージェント方策 π_{-k} のもとで以下が成り立つ[1]．

$$\nabla_{\theta_k} \log p_l(\tau_m|\{\theta_k\}) = \sum_{i \in \Phi_l(k)} \lim_{T\to\infty} \sum_{t=0}^{T} \sum_{s_{0:t}, \boldsymbol{a}_{0:t}} p(s_{0:t}, \boldsymbol{a}_{0:t}) \left(\nabla_{\theta_k} Q_i(s_t, a_t) - \nabla_{\theta_k} V_i(s_t)\right) \tag{7.18}$$

$$= \sum_{i \in \Phi_l(k)} \left(\overline{\boldsymbol{f}}_{i,\boldsymbol{\pi}^{\mathrm{E}}} - \overline{\boldsymbol{f}}_{i,\boldsymbol{\pi}_l}\right) \tag{7.19}$$

$$= \overline{\boldsymbol{f}}_{k,\boldsymbol{\pi}^{\mathrm{E}}} - \overline{\boldsymbol{f}}_{k,\boldsymbol{\pi}_l} \tag{7.20}$$

最後に，提案法のアルゴリズムを Algorithm 5 示す．HeteroSoftQ$_l$ は，メンバーシップ l のもとで Heterogeneous Q-learning[Šošić 17] の価値関数に式 (2.13) を用いたときの最適方策であり，Algorithm 6 にそのアルゴリズムを示す．$L(k)$ はグループ k への割当てが存在するメンバーシップの集合を表す．

7.2.3 計算量評価

エージェント数 N，グループ数 K に対する提案法の実行計算量を評価する．最も計算量の多いのは Algorithm 5 の 10 行目における Inner Loop であり，この Inner Loop を各 Iteration で $K^{N+1} - K(K-1)^N$ 通り実行する必要がある．そのため，1 [iteration] あたりの Inner Loop の実行回数は，座標降下法を用いて更新する場合は $O(K^N)$，6 章で提案した並列化を用いた場合は十分な計算資源のもとで $O(1)$ となる．ただし本提案では，実験に用いる計算機の性能に合わせるため，Algorithm 5 に示すように，各グループ k ごとに $K^N - (K-1)^N$ 通りの Inner Loop のみ並列化することで $O(K)$ とした．並列座標降下法により，全てを並列化した場合の学習速度評価は今後の課題とする．

7.2.4 提案法の立ち位置

逆強化学習において，報酬が軌跡間で異なる場合を扱う研究は二つある．

[1] 文献 [Ziebart 10a] の Theorem 6.4 を inifnite-horizon に拡張した．

Algorithm 5 Estimation for membership and reward

Input: $\mathcal{D}^{\mathrm{E}}$, Feature vectors $\{\boldsymbol{f}_k\}$, #Group K

Output: Reward and miximing weights $\{\boldsymbol{\theta}_k\}, \{w_l\}$

Initialisation

$\theta_k \leftarrow \mathbf{0} \quad \forall k, \quad w_l \leftarrow \frac{1}{L} \quad \forall l$

$\overline{f}_{k,\boldsymbol{\pi}^{\mathrm{E}}} \leftarrow \frac{1}{M} \sum_{i \in \Phi_l(k)} \sum_{(s_t, a_{it}) \sim \tau_m^{\mathrm{E}}} \gamma^t f_i(s_t, a_{it}) \quad \forall k$

for $k = 1, \ldots, K$ **do**

    $\pi_{kl} \leftarrow \mathrm{HeteroSoftQ}_l(\theta_k, \pi_{-kl}) \quad \forall l \in L(k)$

end for

Estimation

for Iteration $= 1, 2, \ldots$ **do**

    Update Eq. (7.12), Eq. (7.15), Eq. (7.13) $\forall l$

    for $k = 1, \ldots, K$ **do**

        Update Eq. (7.14) for k

        $\pi_{kl} \leftarrow \mathrm{HeteroSoftQ}_l(\theta_k, \pi_{-kl}) \quad \forall l \in L(k)$ ▷ Parallel execution

        Sample $\mathcal{D}_l^{\mathrm{sample}}$ from $\boldsymbol{\pi}_l$

        $\mathcal{D}_l^{\mathrm{IS}} \leftarrow \mathcal{D}^{\mathrm{E}} \cup \mathcal{D}_l^{\mathrm{sample}}$

    end for

end for

Algorithm 6 Heterogeneous Soft Q-learning for group k in membership l

Input: Set of agent $\mathcal{N}_k$ (belonging group k), Reward weight $\boldsymbol{\theta}_k$, Other agent's policy π_{-kl}, Entropy coefficient α, Temperature for exploring

for episode $= 1, 2, \cdots$ **do**

    while reach terminal state or max step **do**

        Sepearte exploring and greedy agents according to the current temperature

        Select actions for all agents

        Collect sample $\{(s_{i,t}, \boldsymbol{a}_{i,t}, s_{i,t+1})\}_{i \in \mathcal{N}_k}$ and update Q_k

        Decrease temperature and learning rate

    end while

end for

Babes ら [Babes 11] は，単一エージェント環境において，軌跡を生成した分布が各軌跡で異なる場合の逆強化学習を提案している．直接観測できない軌跡ごとの軌跡分布への割当ては潜在変数として扱われ，軌跡は複数の軌跡分布を重みづけした混合分布から生成されたとみなす．そして，EM アルゴリズムを用いることで，割当てと報酬を同時に推定している．一方，本提案は，この手法のマルチエージェント系への拡張といえる．

Lee らは，逆強化学習による Agent-based model の構築法 [Lee 17] を提案している．この手法では，軌跡のクラスタリングと報酬の推定をそれぞれ別に行う．軌跡のクラスタリングでは，階層クラスタ分析を用いて，軌跡の特徴期待値に基づき軌跡を分類している．次に，クラスタリングされた軌跡が同一のマルコフ決定過程上の軌跡であるとみなし，その状態遷移確率を，軌跡に含まれる状態遷移とドメイン知識から計算する．そして，各クラスタごとに単一エージェント系の逆強化学習を適用することで，報酬を推定している．一方，本提案はマルコフゲーム上でクラスタリングと報酬推定を EM アルゴリズムによって同時推定する点で大きく異なる．

また，エージェントのグループと類似の提案として，Swarm IRL[Šošić 17] がある．Swarm IRL では，不完全観測環境において，観測能力や報酬，方策が同じエージェントからなる swarMDP を仮定している．報酬推定では，swarMDP の性質によりエージェント間で価値関数が同じであり，十分な時刻経過のもとで観測確率が定常となることを利用して，単一エージェントの問題とみなしている．一方，本提案は，完全観測環境で複数のグループが存在する点で異なる．ただし，提案法の方策更新時には 6 章で用いた座標降下法のように一つのグループのみ更新する．そのため，グループごとの更新については，Swarm IRL と類似の手順を用いる．

7.3 計算機実験

7.3.1 実験設定

Three agents grid navigation

環境には，エージェント 3 体のうち 2 体と 1 体で異なる報酬をもつ Grid navigation を用いる．図 7.2 に初期状態と状態遷移を示す．この環境におけるエージェントの目標は，3×3 のグリッド上で図 7.2 に示す初期配置から，エージェント 1,2 は G_1，エージェント 3 は G_2 からそれぞれゴール座標 G に向かうことである．エージェントの観測はエージェント i の座標 x_i を全エージェントについて組み合わせた $\boldsymbol{x}$，行動は (g, d) であり，g は目指す座標（G_1 か G_2），d は回転方向（時計回りか反時計回り）をそれぞれ表す．状態遷移は確率的であり，同じセルへ移動しようとしたエージェントがいた場合，ランダムに選ばれたエージェントだけが遷移できる．全エージェントがゴール座標 G に到達したとき，任意の行動でエピソードを打ち切る．

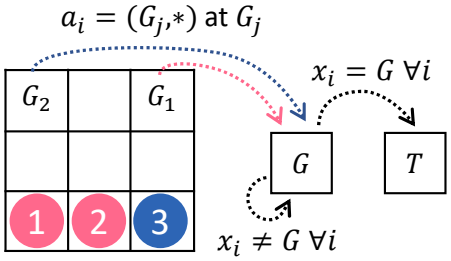


図 7.2: 環境の初期状態と状態遷移（円はエージェント，数字はエージェント番号を表す．各エージェントは，座標 G_j で行動 $(G_j, *)$ を選択するとゴール座標 G へ遷移し，全エージェントが G にいるとき任意の行動で終端状態 T へ遷移する．）

エキスパート軌跡

エキスパート軌跡には，図 7.3 に示す二種類のエキスパート軌跡を用いた．図 7.3(a) に示す軌跡は，外周セルを反時計回りでゴール座標まで移動する軌跡であり，全エージェントが最短ステップでゴールに到達できる．図 7.3(b) に示す軌跡は時計回りで移動するため，エージェント 2 が 1 ステップ余分に移動する準最適な軌跡である．各軌跡は，エキスパート報酬のもと強化学習で得られた方策から最大ステップ 10 として 100 本サンプルした．エキスパート報酬は，ゴール座標 G における報酬を 50 とし，それ以外のセルではエキスパート通りの行動をとったとき 20，逸脱した場合は -20 とした．学習では最大エピソード数 1000，最大ステップ数 30，エントロピーの係数 0.1，割引率 0.9 として Heterogeneous Soft Q-learning で定常方策を獲得する．学習中は，更新するグループ以外の方策は固定されており，全てのグループを 10 回ずつ更新するまで 1 グループずつ順番に学習する．

(a) 反時計回り　　(b) 時計回り

図 7.3: エキスパート軌跡

パラメータ

割引率 γ は 0.9，エントロピーの係数 α はエキスパート方策が決定的方策であることから 0.1 とする．報酬の更新には，0.01 で重みづけされた L2 ノルムを含む勾配をステップサイズ $\beta = 0.005$ で更新する．特徴ベクトル $\boldsymbol{f}_i$ は状態 s とエージェントごとの行動 a_i を組み合わせたバイナリベクトルとする．勾配計算に用いられる軌跡には，最大ステップ数 10 とし

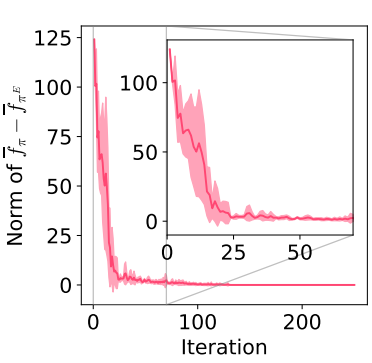
て 1000 本のサンプルで計算する．Inner Loop には，最大エピソード数 1000，最大ステップ数 30 の Heterogeneous Soft Q-learning を用いる．

7.3.2　実験結果

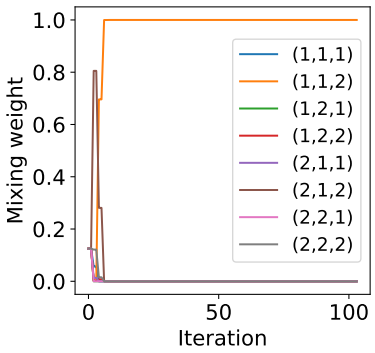
推定するグループ数 $K = 2$ の場合

推定するグループ数を $K = 2$ とした場合の結果（10 試行）を示す．図 7.4，図 7.5 に重み $\boldsymbol{\theta}_k$ の勾配のノルム推移，メンバーシップの混合係数の推移例，エキスパート報酬の獲得量を，表 7.1 に収束率，収束時の繰り返し数と推定されたメンバーシップについて，各エキスパートに対する推定結果をそれぞれ示す．混合係数のメンバーシップ l は $(\phi_l(1), \phi_l(2), \phi_l(3))$ で表され，例えば図 7.4(b) の推定結果 (1,2,2) はエージェント 1 はグループ 1，その他は 2 に属することを表す．

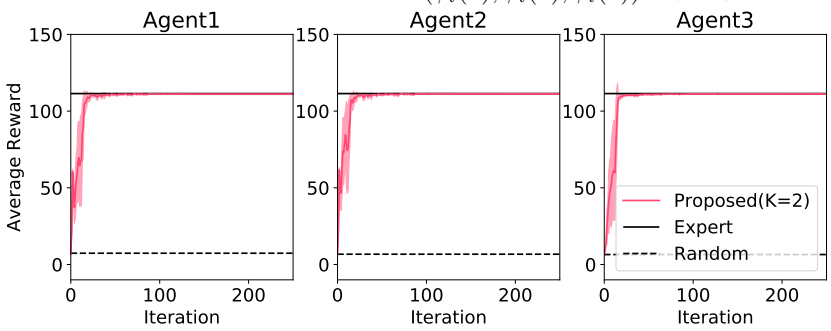
表 7.1 の収束率から，いずれも推定方策がエキスパートに一致したといえる．また，推定されたメンバーシップは，全ての試行において，エージェント 1,2 を同一グループに割当てられており，エキスパートと同一の結果が得られた．



(a) 勾配の推移

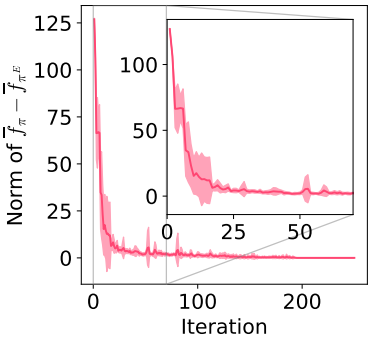


(b) 混合係数の推移例（ラベルはメンバーシップ $(\phi_l(1), \phi_l(2), \phi_l(3))$ を表す）

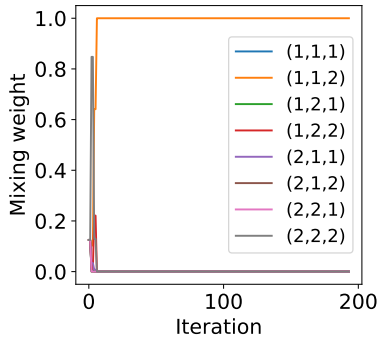


(c) 報酬獲得量の推移

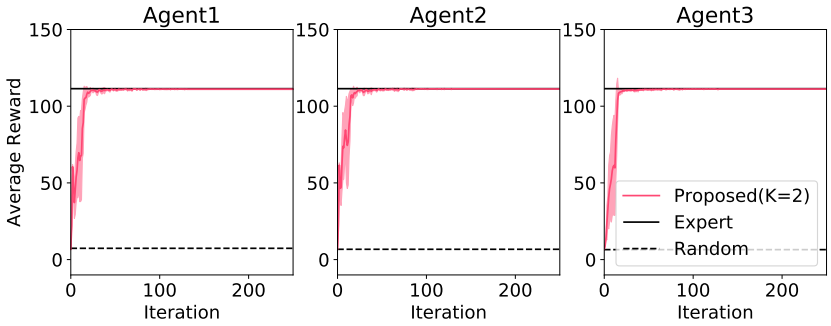
図 7.4: 推定グループ数 $K = 2$ かつ反時計回りの場合の各推定値の推移



(a) 勾配の推移



(b) 混合係数の推移例（ラベルはメンバーシップ $(\phi_l(1), \phi_l(2), \phi_l(3))$ を表す）



(c) 報酬獲得量の推移

図 7.5: 推定グループ数 $K = 2$ かつ時計回りの場合の各推定値の推移

表 7.1: 推定グループ数 $K = 2$ の場合の各エキスパート軌跡に対する収束率と推定されたメンバーシップ（Conv. Rate は 10 試行のうち収束した割合，#Iteration は収束時の繰り返し数，メンバーシップの結果 $(\phi_l(1), \phi_l(2), \phi_l(3)) : x$ は，10 試行のうちエージェントの属するグループのインデックスの組 $(\phi_l(1), \phi_l(2), \phi_l(3))$ に対し，その混合係数が最大になった試行が x 回あったことを示す.）

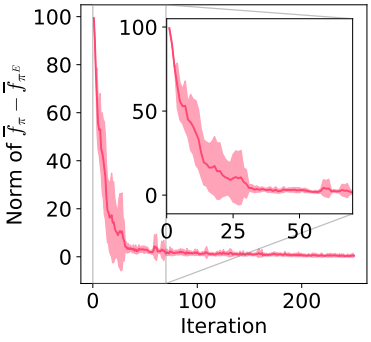
Expert	Conv. Rate [%]	#Iteration	Estimated membership
counter-clockwise	100.0	90.8 ± 23.7	(1, 1, 2):5,(2, 2, 1):5,
clockwise	100.0	138.6 ± 37.7	(1, 1, 2):4,(2, 2, 1):6,

推定するグループ数 $K = 3$ の場合

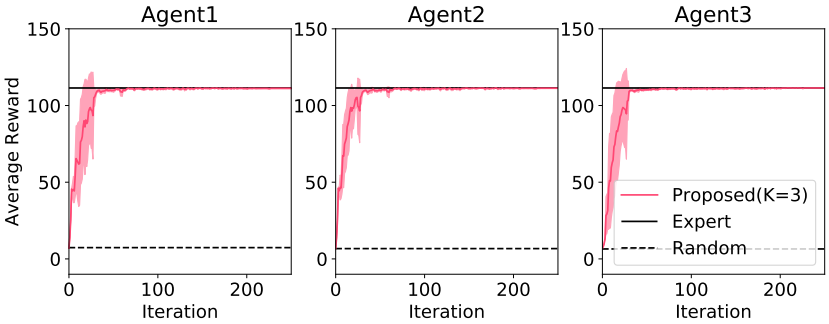
推定するグループ数を $K = 3$ とした場合の結果（10 試行）を示す．図 7.6，図 7.7，表 7.2 に，各エキパートに対する推定結果をそれぞれ示す．

表 7.2 の結果から，$K = 2$ の場合と比べ収束率が下がったものの，図 7.6，図 7.7 の結果から，特徴期待値の差や獲得報酬はエキスパートに漸近していることを示している．

ただし，表 7.2 に示した反時計回りの結果から，エキスパートと異なるメンバーシップが推定される場合があった．



(a) 勾配の推移

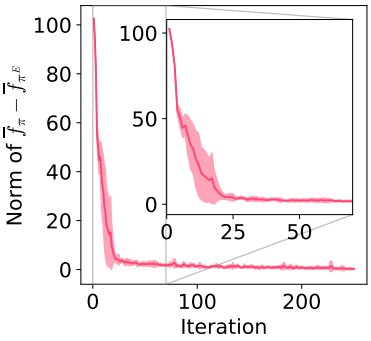


(b) 報酬獲得量の推移

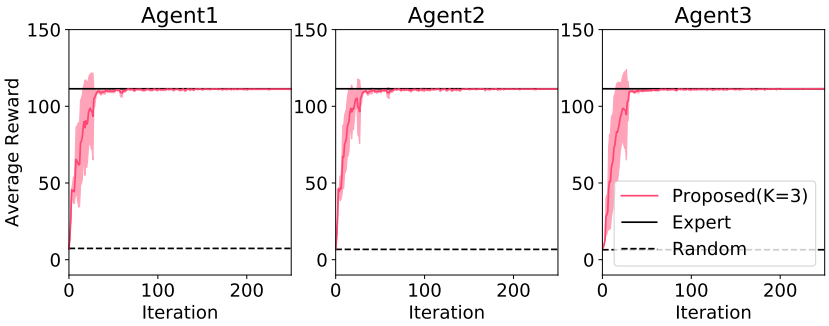
図 7.6: 推定グループ数 $K = 3$ かつ反時計回りの場合の各推定値の推移

表 7.2: 推定グループ数 $K = 3$ の場合の各エキスパート軌跡に対する収束率と推定されたメンバーシップ（Conv. Rate は 10 試行のうち収束した割合，#Iteration は収束時の繰り返し数，メンバーシップの結果 $(\phi_l(1), \phi_l(2), \phi_l(3)) : x$ は，10 試行のうちエージェントの属するグループのインデックスの組 $(\phi_l(1), \phi_l(2), \phi_l(3))$ に対し，その混合係数が最大になった試行が x 回あったことを示す．）

Expert	Conv. Rate [%]	#Iteration	Estimated membership
counter-clockwise	60.0	184.3 ± 50.6	(1, 1, 2):1,(1, 1, 3):1,(1, 2, 3):2,(2, 2, 1):2, (2, 2, 3):1,(2, 3, 1):2,(3, 3, 1):1,
clockwise	70.0	209.3 ± 29.2	(1, 1, 2):3,(1, 1, 3):2,(2, 2, 1):4,(3, 3, 2):1,



(a) 勾配の推移



(b) 報酬獲得量の推移

図 7.7: 推定グループ数 $K = 3$ かつ時計回りの場合の各推定値の推移

7.4 考察

提案法の収束率は，エキスパートのグループ数と同じ $K = 2$ の場合は全て収束したが，エキスパートのグループ数より多い $K = 3$ の場合は減少した．この結果は，メンバーシップの数が $K = 2$ のとき 8，$K = 3$ のとき 27 と探索空間が大きくなる点で妥当といえる．実際，表 7.1，表 7.2 の収束までの繰り返し数を比較すると，2 倍程度の差が生じている．一方で，提案法により推定された方策は，推定するグループ数 K によらずエキスパートと同等であった．この結果は，エキスパート軌跡が決定的な軌跡であるのに対し，推定方策がボルツマン分布で表されるため，特徴期待値の差が十分に小さくなるために多くの更新をを必要とすることに原因があると考えられる．よって，探索するメンバーシップの空間に併せて，繰り返し数や閾値を調整することで，収束率は改善の余地がある．

一方，推定されるメンバーシップは，エキスパートのグループ数より多い $K = 3$ の場合に全てのエージェントが異なるグループに割り当てられる結果が含まれていた．この結果は，提案法の目的がエキスパート軌跡の尤度を最大化のため，グループが異なっていても同一報酬を推定すればよいことから，妥当な結果といる．よって，必要最小限のグループ数の推定を必要とする場合には，K を調整することで推定可能なグループ数を判断するか，推定時に予めグループ数を最小化するための罰則項を導入するなど，何らかの工夫が必要といえる．

7.5 結言

本章では，報酬と方策が同一のグループが複数あるマルチエージェント系において，軌跡からエージェントのグループへのメンバーシップと報酬を同時推定する方法を提案した．既存法と比較して，提案法はマルコフゲームを対象に複数グループを仮定する点で異なる．計算機実験では，エージェント 3 体の Grid navigation を用いて，2 種類の報酬があるエージェントの軌跡に対し，妥当な解を推定できることを示した．また，必要最小限のグループ数による推定を必要とする場合は，提案法が過剰な分割を起こす可能性があることから，何らかの工夫を必要とすることを示した．

第8章　結論

本論文では，マルチエージェント系（MAS）における Agent-Based Model（ABM）を用いた現実の振舞いの予測，もしくはマルチエージェント強化学習（MARL）を用いた望ましい振舞いへの制御を対象に，エージェント間の個体差を前提として，エージェントの行動則や，その報酬関数をエージェントの軌跡から自動推定する手法について提案した．対象問題は,「群衆 Agent-based Model における異種戦略推定」，マルチエージェント逆強化学習における「並列化による報酬の学習速度改善」,「グループ構造と報酬の同時推定」の三つからなる．

群衆 Agent-based Model における異種戦略推定

ABM の実施には，妥当なエージェントの行動則を設計する必要があり，人手による試行錯誤的な設計ではなく，人手の補助，もしくは自動的な行動則の設計法が必要とされてきた．そこで，行動則の自動設計例として，エージェントが「どの状態を目標とするか」という意思決定の基準である戦略推定を対象とした．

既存研究では，再現したい振舞いを示す軌跡から行動則を自動設計する手法として，進化計算を用いた手法が提案されている．しかし，この既存法では同種な行動則しか扱えず，多様な振舞いを実現するのに十分ではなかった．そこで，Automatically Defined Groups[Hara 99, 原 00] を導入することで，同一の戦略をグループ化する異種戦略の推定法を提案した．

計算機実験では，群衆シミュレーションを対象に，人工的に生成した軌跡から戦略を推定した．その結果，既存法では推定が困難な軌跡に対して，提案法により推定可能なことを確かめた．また，戦略が異なっていても観測される軌跡が一致する場合があることを示し，多様な戦略を獲得できる一方で，人の持つ真の戦略を特定することへの限界を示した．加えて，実際の軌跡へ適用する場合の課題として，推定誤差の緩和，もしくは人の観測する特徴量の設計が重要であることを示した．

並列化による報酬の学習速度改善

MAS において望ましい振舞いへエージェントを制御するマルチエージェント強化学習では，報酬設計の困難性が従来から指摘されていた．そこで本論文は，この問題に対するアプローチであるマルチエージェント逆強化学習を対象に，個々のエージェントの報酬を巡回更

新するのではなく，全てのエージェントの報酬を並列的に更新する並列座標降下法を用いた推定法を提案した．提案法は，エージェント数に対して探索空間が指数関数的に増大する処理を，エージェント数の数だけ何度も更新することなく，並列計算できる利点がある．

計算機実験では，エージェント 2 体の Grid navigation を対象に，特定の Nash 均衡解を示す軌跡から報酬を推定した．結果から，単純に並列化しても学習速度が改善しない場合があり，この原因は，各エージェントのローカルな処理で必要となる他エージェントの方策が与えられた軌跡を生成した方策と異なり，その差異がボトルネックになっていることを確認した．このボトルネックに対しては，他エージェントの方策を疑似方策に代替することによって，同期による待ち時間を解消し，並列化による学習速度改善効果が見込めることも示した．

グループ構造と報酬の同時推定

マルチエージェント逆強化学習におけるもう一つの課題として，推定する報酬構造への仮定がある．前述の研究を含め，従来のマルチエージェント逆強化学習は，zero-sum，全エージェントで共通，個々に異なる報酬関数など，推定する報酬構造を仮定する必要があった．しかし，このような仮定を軌跡から直接立てるのは，時系列性や不確実性を扱う必要がある点で容易ではない．一方で，個々の報酬を求めるのは冗長な表現となる．

そこで，報酬と方策が同一なエージェントのグループを仮定し，EM アルゴリズムをマルチエージェント逆強化学習に導入することで，グループ構造と報酬の同時推定法を提案した．

計算機実験では，Grid navigation をエージェント 3 体に拡張した環境を対象に，軌跡のみから妥当なグループ構造と報酬を推定できることを示した．また，必要最小限のグループ数による推定を必要とする場合は，提案法が過剰な分割を起こす可能性があることから，何らかの工夫を必要とすることを実験的に示した．

今後の課題は，6 章や 7 章で提案したマルチエージェント逆強化学習を，5 章で用いた群衆避難モデルへ適用することである．逆強化学習は，強化学習問題の報酬設計に対するアプローチである一方，軌跡に含まれるエージェントの意図を報酬として推定する用途にも用いられる．そのため，ABM において実際の軌跡データに適用することで，5 章で考慮していないエージェントの意思決定における時系列性や不確実性を扱える可能性がある．

ただし ABM にマルチエージェント逆強化学習を適用するには，まず，不完全知覚環境への拡張を必要とする．図 4.1 に示したように，提案法を含め，マルチエージェント逆強化学習は完全観測環境を対象しているが，ABM は不完全知覚を扱うことが多い．この課題に対しては，不完全観測を扱う逆強化学習 [Choi 11] のマルチエージェント系への拡張が考えられる．次に，エージェント数へのスケール性に取り組む必要がある．並列座標降下法を用いた提案では，並列化自体が有効に働くか検証するため，エージェント 2 体の場合を対象とした．エージェント 3 体以上の場合は同期的な方策交換がボトルネックとなる可能性があることから，非同期的な並列化への拡張が必要と考えられる．また，EM アルゴリズムを用いた

提案は，グループ構造の組合せの数だけ強化学習問題を扱う必要があり，例え前述の並列計算を用いても計算量の増大に追いつかない．そのため，推定する過程で尤度の低いグループ構造を枝刈りするといった工夫が考えられる．

最後に，すべての提案に共通する課題として，推定結果の転移可能性がある．本論文の提案は与えられた軌跡を再現するものであり，軌跡に含まれない状況での振舞いを保証していない．その解決法として，複数の環境を考慮するメタ学習や，環境に依存しない特徴量の自動設計への拡張が考えられる．

参考文献

[Arora 18] Arora, S. and Doshi, P.: A Survey of Inverse Reinforcement Learning: Challenges, Methods and Progress (2018)

[Babes 11] Babes, M., Marivate, V. N., Subramanian, K., and Littman, M. L.: Apprenticeship Learning About Multiple Intentions, in *ICML*, pp. 897–904 (2011)

[Bhattacharyya 18] Bhattacharyya, R. P., Phillips, D. J., Wulfe, B., Morton, J., Kuefler, A., and Kochenderfer, M. J.: Multi-Agent Imitation Learning for Driving Simulation, in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1534–1539 (2018)

[Bhattacharyya 19] Bhattacharyya, R. P., Phillips, D. J., Liu, C., Gupta, J. K., Driggs-Campbell, K., and Kochenderfer, M. J.: Simulating Emergent Properties of Human Driving Behavior Using Multi-Agent Reward Augmented Imitation Learning, in *2019 International Conference on Robotics and Automation (ICRA)*, pp. 789–795 (2019)

[Bishop 08] Bishop, C. M.: パターン認識と機械学習下, ベイズ理論による統計的予測 (2008)

[Bogert 14] Bogert, K. and Doshi, P.: Multi-robot Inverse Reinforcement Learning Under Occlusion with Interactions, in *Proceedings of the 2014 International Conference on Autonomous Agents and Multi-agent Systems*, AAMAS '14, pp. 173–180 (2014)

[Borges 10] Borges, C. E., Alonso, C. L., and Montaña, J. L.: Model Selection in Genetic Programming, in *Proceedings of the 12th Annual Conference on Genetic and Evolutionary Computation*, GECCO '10, p. 985986 (2010)

[BoussaïD 13] BoussaïD, I., Lepagnot, J., and Siarry, P.: A survey on optimization metaheuristics, *Information sciences*, Vol. 237, pp. 82–117 (2013)

[Bowling 03] Bowling, M.: Multiagent learning in the presence of agents with limitations, Technical report, CARNEGIE-MELLON UNIV PITTSBURGH PA SCHOOL OF COMPUTER SCIENCE (2003)

[Choi 11] Choi, J. and Kim, K.-E.: Inverse Reinforcement Learning in Partially Observable Environments, *J. Mach. Learn. Res.*, Vol. 12, No. null, p. 691730 (2011)

[Dorri 18] Dorri, A., Kanhere, S. S., and Jurdak, R.: Multi-Agent Systems: A Survey, *IEEE Access*, Vol. 6, pp. 28573–28593 (2018)

[Eiben 15] Eiben, A. E. and Smith, J.: From evolutionary computation to the evolution of things, *Nature*, Vol. 521, No. 7553, pp. 476–482 (2015)

[Ferreira 01] Ferreira, C.: Gene expression programming: a new adaptive algorithm for solving problems, *arXiv preprint cs/0102027* (2001)

[Finn 16] Finn, C., Levine, S., and Abbeel, P.: Guided cost learning: Deep inverse optimal control via policy optimization, in *International conference on machine learning*, pp. 49–58 (2016)

[Haarnoja 17] Haarnoja, T., Tang, H., Abbeel, P., and Levine, S.: Reinforcement Learning with Deep Energy-based Policies, in *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, pp. 1352–1361, JMLR.org (2017)

[Hara 99] Hara, A. and Nagao, T.: Emergence of the cooperative behavior using ADG; Automatically Defined Groups., in *GECCO*, pp. 1039–1046 (1999)

[Hara 05] Hara, A., Ichimura, T., and Yoshida, K.: Discovering multiple diagnostic rules from coronary heart disease database using automatically defined groups, *Journal of Intelligent Manufacturing*, Vol. 16, No. 6, pp. 645–661 (2005)

[Heath 09] Heath, B., Hill, R., and Ciarallo, F.: A survey of agent-based modeling practices (January 1998 to July 2008), *Journal of Artificial Societies and Social Simulation*, Vol. 12, No. 4, p. 9 (2009)

[Helbing 00] Helbing, D., Farkas, I., and Vicsek, T.: Simulating dynamical features of escape panic, *Nature*, Vol. 407, No. 6803, pp. 487–490 (2000)

[Helbing 07] Helbing, D., Johansson, A., and Al-Abideen, H. Z.: Dynamics of crowd disasters: An empirical study, *Physical review E*, Vol. 75, No. 4, p. 046109 (2007)

[Hernandez-Leal 19] Hernandez-Leal, P., Kartal, B., and Taylor, M. E.: A survey and critique of multiagent deep reinforcement learning, *Autonomous Agents and Multi-Agent Systems*, Vol. 33, No. 6, pp. 750–797 (2019)

[Hu 03] Hu, J. and Wellman, M. P.: Nash Q-learning for general-sum stochastic games, *Journal of machine learning research*, Vol. 4, No. Nov, pp. 1039–1069 (2003)

[Jensen 17] Jensen, T. and J.L. Chappin, mile : Automating agent-based modeling: Data-driven generation and application of innovation diffusion models, *Environmental Modelling & Software*, Vol. 92, pp. 261 – 268 (2017)

[Jeon 20] Jeon, W., Barde, P., Nowrouzezahrai, D., and Pineau, J.: Scalable and sample-efficient multi-agent imitation learning, in *Proceedings of the Workshop on Artificial Intelligence Safety, co-located with 34th AAAI Conference on Artificial Intelligence, SafeAI@ AAAI* (2020)

[Kavak 18] Kavak, H., Padilla, J. J., Lynch, C. J., and Diallo, S. Y.: Big Data, Agents, and Machine Learning: Towards a Data-Driven Agent-Based Modeling Approach, in *Proceedings of the Annual Simulation Symposium*, ANSS '18, San Diego, CA, USA (2018), Society for Computer Simulation International

[Keller 19] Keller, N. and Hu, X.: Towards Data-Driven Simulation Modeling for Mobile Agent-Based Systems, *ACM Trans. Model. Comput. Simul.*, Vol. 29, No. 1 (2019)

[Le 16] Le, N., Xuan, H. N., Brabazon, A., and Thi, T. P.: Complexity measures in Genetic Programming learning: A brief review, in *2016 IEEE Congress on Evolutionary Computation (CEC)*, pp. 2409–2416 (2016)

[Le 17] Le, H. M., Yue, Y., Carr, P., and Lucey, P.: Coordinated Multi-Agent Imitation Learning, in Precup, D. and Teh, Y. W. eds., *Proceedings of the 34th International Conference on Machine Learning*, Vol. 70 of *Proceedings of Machine Learning Research*, pp. 1995–2003, International Convention Centre, Sydney, Australia (2017), PMLR

[Lee 17] Lee, K., Rucker, M., Scherer, W. T., Beling, P. A., Gerber, M. S., and Kang, H.: Agent-based model construction using inverse reinforcement learning, in *2017 Winter Simulation Conference (WSC)*, pp. 1264–1275IEEE (2017)

[Lerner 07] Lerner, A., Chrysanthou, Y., and Lischinski, D.: Crowds by Example, *Computer Graphics Forum*, Vol. 26, No. 3, pp. 655–664 (2007)

[Lin 18] Lin, X., Beling, P. A., and Cogill, R.: Multiagent Inverse Reinforcement Learning for Two-Person Zero-Sum Games, *IEEE Transactions on Games*, Vol. 10, No. 1, pp. 56–68 (2018)

[Macal 18] Macal, C. M.: TUTORIAL ON AGENT-BASED MODELING AND SIMULATION: ABM DESIGN FOR THE ZOMBIE APOCALYPSE, in *2018 Winter Simulation Conference (WSC)*, pp. 207–221 (2018)

[Melnikov 16] Melnikov, V., Krzhizhanovskaya, V., Lees, M., and Boukhanovsky, A.: Data-driven Travel Demand Modelling and Agent-based Traffic Simulation in Amsterdam Urban Area, *Procedia Computer Science*, Vol. 80, pp. 2030 – 2041 (2016), International Conference on Computational Science 2016, ICCS 2016, 6-8 June 2016, San Diego, California, USA

[Mguni 19] Mguni, D., Jennings, J., Sison, E., Valcarcel Macua, S., Ceppi, S., and Cote, Munoz de E.: Coordinating the Crowd: Inducing Desirable Equilibria in Non-Cooperative Systems, in *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems*, AAMAS '19, p. 386394, Richland, SC (2019), International Foundation for Autonomous Agents and Multiagent Systems

[Montaña 11] Montaña, J. L., Alonso, C. L., Borges, C. E., and Dehesa, de la J.: Penalty Functions for Genetic Programming Algorithms, in Murgante, B., Gervasi, O., Iglesias, A., Taniar, D., and Apduhan, B. O. eds., *Computational Science and Its Applications - ICCSA 2011*, pp. 550–562, Berlin, Heidelberg (2011), Springer Berlin Heidelberg

[Natarajan 10] Natarajan, S., Kunapuli, G., Judah, K., Tadepalli, P., Kersting, K., and Shavlik, J.: Multi-Agent Inverse Reinforcement Learning, in *2010 Ninth International Conference on Machine Learning and Applications*, pp. 395–400 (2010)

[Ng 00] Ng, A. Y. and Russell, S. J.: Algorithms for Inverse Reinforcement Learning, in *Proceedings of the Seventeenth International Conference on Machine Learning*, ICML '00, pp. 663–670 (2000)

[Ogino 04] Ogino, S. and Nagao, T.: The chaos analysis of long memory process in artificial stock markets consist of multi-agents, in *2004 International Conference on Cyberworlds*, pp. 249–253IEEE (2004)

[Panait 05] Panait, L. and Luke, S.: Cooperative multi-agent learning: The state of the art, *Autonomous agents and multi-agent systems*, Vol. 11, No. 3, pp. 387–434 (2005)

[Ramachandran 07] Ramachandran, D. and Amir, E.: Bayesian Inverse Reinforcement Learning, in *IJCAI* (2007)

[Ratliff 19] Ratliff, L. J., Dong, R., Sekar, S., and Fiez, T.: A Perspective on Incentive Design: Challenges and Opportunities, *Annual Review of Control, Robotics, and Autonomous Systems*, Vol. 2, pp. 305–338 (2019)

[Reddy 12] Reddy, T. S., Gopikrishna, V., Zaruba, G., and Huber, M.: Inverse reinforcement learning for decentralized non-cooperative multiagent systems, in *2012 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pp. 1930–1935, Seoul, Korea (South) (2012)

[Russell 98] Russell, S.: Learning agents for uncertain environments, in *Proceedings of the eleventh annual conference on Computational learning theory*, pp. 101–103 (1998)

[Sajjad 16] Sajjad, M., Singh, K., Paik, E., and Ahn, C.-W.: A data-driven approach for agent-based modeling: simulating the dynamics of family formation, *Journal of Artificial Societies and Social Simulation*, Vol. 19, No. 1, p. 9 (2016)

[Segaran 08] Segaran, T.: 集合知プログラミング, オライリージャパン (2008)

[SFModelSup] Panic: A quantitative analysis: Simulating dynamical features of pedestrian escape panic, `http://angel.elte.hu/panic/` [accessed 2015/10/11]

[Sigaud 10] Sigaud, O. and Buffet, O.: *Markov Decision Processes in Artificial Intelligence*, Wiley-IEEE Press (2010)

[Song 18] Song, J., Ren, H., Sadigh, D., and Ermon, S.: Multi-Agent Generative Adversarial Imitation Learning, in Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R. eds., *Advances in Neural Information Processing Systems*, Vol. 31, pp. 7461–7472, Curran Associates, Inc. (2018)

[Šošić 17] Šošić, A., KhudaBukhsh, W. R., Zoubir, A. M., and Koeppl, H.: Inverse Reinforcement Learning in Swarm Systems, in *Proceedings of the 16th Conference on Autonomous Agents and MultiAgent Systems*, AAMAS '17, pp. 1413–1421, International Foundation for Autonomous Agents and Multiagent Systems (2017)

[Sutton 18] Sutton, R. S. and Barto, A. G.: *Reinforcement Learning: An Introduction*, A Bradford Book, Cambridge, MA, USA (2018)

[Truong 16] Truong, T. M., Amblard, F., Gaudou, B., and Blanc, C. S.: CFBM-A Framework for Data Driven Approach in Agent-Based Modeling and Simulation, in *International Conference on Nature of Computation and Communication*, pp. 264–275Springer (2016)

[Wang 18] Wang, X. and Klabjan, D.: Competitive Multi-agent Inverse Reinforcement Learning with Sub-optimal Demonstrations, in *Proceedings of the 35th International Conference on Machine Learning*, Vol. 80 of *Proceedings of Machine Learning Research*, pp. 5143–5151, PMLR (2018)

[Wang 21] Wang, X., Ning, Z., and Guo, S.: Multi-Agent Imitation Learning for Pervasive Edge Computing: A Decentralized Computation Offloading Algorithm, *IEEE Transactions on Parallel and Distributed Systems*, Vol. 32, No. 2, pp. 411–425 (2021)

[Wei 19] Wei, E., Wicke, D., and Luke, S.: Multiagent Adversarial Inverse Reinforcement Learning, in *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems*, pp. 2265–2266 (2019)

[Wright 15] Wright, S. J.: Coordinate descent algorithms, *Mathematical Programming*, Vol. 151, No. 1, pp. 3–34 (2015)

[Wu 19] Wu, G., Li, Y., and Luo, J.: Transforming Policy via Reward Advancement, in *2019 IEEE 58th Conference on Decision and Control (CDC)*, pp. 4609–4614 (2019)

[Yamaguchi 11] Yamaguchi, K., Berg, A. C., Ortiz, L. E., and Berg, T. L.: Who are you with and where are you going?, in *CVPR 2011*, pp. 1345–1352 (2011)

[Yang 05] Yang, L., Zhao, D., Li, J., and Fang, T.: Simulation of the kin behavior in building occupant evacuation based on cellular automaton, *Building and Environment*, Vol. 40, No. 3, pp. 411–415 (2005)

[Yang 18] Yang, J., Ye, X., Trivedi, R., Xu, H., and Zha, H.: Deep Mean Field Games for Learning Optimal Behavior Policy of Large Populations, in *International Conference on Learning Representations* (2018)

[Yang 20a] Yang, F., Vereshchaka, A., Chen, C., and Dong, W.: Bayesian Multi-type Mean Field Multi-agent Imitation Learning, *Advances in Neural Information Processing Systems*, Vol. 33, (2020)

[Yang 20b] Yang, M., Li, Y., Zhou, X., Lu, H., Tian, Z., and Luo, J.: Inferring Passengers ' Interactive Choices on Public Transits via MA-AL: Multi-Agent Apprenticeship Learning, in *Proceedings of The Web Conference 2020*, pp. 1637–1647 (2020)

[Yu 19] Yu, L., Song, J., and Ermon, S.: Multi-Agent Adversarial Inverse Reinforcement Learning, in *International Conference on Machine Learning*, pp. 7194–7201 (2019)

[Zhan 19] Zhan, E., Zheng, S., Yue, Y., Sha, L., and Lucey, P.: Generating Multi-Agent Trajectories using Programmatic Weak Supervision, in *International Conference on Learning Representations* (2019)

[Zhang 15] Zhang, H., Vorobeychik, Y., Letchford, J., and Lakkaraju, K.: Data-Driven Agent-Based Modeling, with Application to Rooftop Solar Adoption, in *Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems*, AAMAS '15, p. 513521, Richland, SC (2015), International Foundation for Autonomous Agents and Multiagent Systems

[Zhifei 12] Zhifei, S. and Er Meng Joo, : A review of inverse reinforcement learning theory and recent advances, in *2012 IEEE Congress on Evolutionary Computation*, pp. 1–8, Brisbane, Australia (2012), IEEE

[Zhong 14] Zhong, J., Luo, L., Cai, W., and Lees, M.: Automatic rule identification for agent-based crowd models through gene expression programming, in *Proceedings of the 2014 international conference on Autonomous agents and multi-agent systems*, pp. 1125–1132 (2014)

[Zhong 15] Zhong, J., Cai, W., Luo, L., and Yin, H.: Learning Behavior Patterns from Video: A Data-Driven Framework for Agent-Based Crowd Modeling, in *Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems*, AAMAS '15, p. 801809, Richland, SC (2015), International Foundation for Autonomous Agents and Multiagent Systems

[Zhong 16] Zhong, J., Ong, Y., and Cai, W.: Self-Learning Gene Expression Programming, *IEEE Transactions on Evolutionary Computation*, Vol. 20, No. 1, pp. 65–80 (2016)

[Zhou 18] Zhou, Z., Bloem, M., and Bambos, N.: Infinite Time Horizon Maximum Causal Entropy Inverse Reinforcement Learning, *IEEE Transactions on Automatic Control*, Vol. 63, No. 9, pp. 2787–2802 (2018)

[Ziebart 08] Ziebart, B. D., Maas, A. L., Bagnell, J. A., and Dey, A. K.: Maximum Entropy Inverse Reinforcement Learning, in *AAAI* (2008)

[Ziebart 10a] Ziebart, B. D.: Modeling purposeful adaptive behavior with the principle of maximum causal entropy (2010)

[Ziebart 10b] Ziebart, B. D., Bagnell, J. A., and Dey, A. K.: Modeling Interaction via the Principle of Maximum Causal Entropy, in *ICML* (2010)

[伊庭 96] 伊庭 斉志：遺伝的プログラミング, 東京電機大学出版局 (1996)

[原 00] 原 章, 長尾 智晴：自動グループ構成手法 ADG によるマルチエージェントの行動制御, 情報処理学会論文誌, Vol. 41, No. 4, pp. 1063–1072 (2000)

[荒井 98] 荒井 幸代, 宮崎 和光, 小林 重信：マルチエージェント強化学習の方法論：Q-Learning と Profit Sharing による接近, 人工知能, Vol. 13, No. 4, pp. 609–618 (1998)

[哲也 93] 哲也 樋口, 宏明 北野：遺伝的アルゴリズムとその応用, 情報処理, Vol. 34, No. 7 (1993)

[浪越 17] 浪越 圭一, 荒井 幸代：人流データに基づく避難者の適応戦略抽出, 人工知能学会全国大会論文集, Vol. JSAI2017, pp. 2M4OS32a5–2M4OS32a5 (2017)

謝辞

本論文の執筆にあたり，様々な方からご支援いただきました．

研究室配属から 7 年間，長きにわたりご指導頂いた荒井幸代先生に感謝いたします．研究者としても，人としても，未熟な私に対し，先生の熱心なご指導なくしては，この論文の完成はありませんでした．

産業技術総合研究所の野田五十樹先生には，研究に対するご助言と，貴重な研究機会を与えて頂きました．ご多忙の中，中々研究成果の出ない私に対し，辛抱強くお付き合いいただきました．感謝いたします．

また，荒井研究室の研究メンバーの方々には，様々な刺激をいただきました．特に，最も長い時間を共に過ごした，北里勇樹さん，石川翔太さん，中田勇介君や，配属当初からの唯一の同期であり，卒業後も私を気にかけていただいた吉永和史氏に，それぞれ感謝いたします．

最後に，私の意思を尊重し見守っていただいた家族へ．ありがとうございました．

研究業績

【学術雑誌に発表した論文】

1. (査読 2 名) 浪越圭一, 荒井幸代 : 進化計算による人流データからの異種戦略抽出, 人工知能学会論文誌, Vol.32, No.5, AG16-D (2017.9)

【国際会議における発表】

2. (口頭発表・査読 5 名) Namikoshi, K. and Arai, S.: Estimation of the Heterogeneous Strategies from Action Log, in *Genetic and Evolutionary Computation Conference 2018*, pp.1310-1317 (2018.7 Japan)
3. (口頭発表・ポスター・査読 2 名) Namikoshi, K. and Arai, S.: Estimation of agent's rewards using multi-agent maximum discounted causal entropy inverse reinforcement learning, in *Adaptive and Learning Agents Workshop 2019* (2019.5 Canada)

【国内学会・シンポジウムにおける発表】

4. (ポスター・査読なし) 浪越圭一, 荒井幸代 : 歩行軌跡に基づく歩行者の行動規範の同定, Joint Agent Workshops & Symposium 2015 (2015.9 石川)
5. (口頭発表・査読なし) 浪越圭一, 荒井幸代 : 避難行動データに基づく非常口選択規範の同定, 情報処理学会第 78 回全国大会 (2016.3 神奈川)
6. (口頭発表・ポスター・査読なし) 浪越圭一, 荒井幸代 : 進化計算による災害時行動データからの避難規範抽出法, 第 30 回人工知能学会全国大会, 1L2-1in1 (2016.5 福岡)
7. (口頭発表・査読なし) 浪越圭一, 荒井幸代 : 人流データに基づく複数の行動決定ルール抽出, 第 11 回データ指向構成マイニングとシミュレーション研究会 (2016.11 神奈川)
8. (口頭発表・ポスター・査読なし) 浪越圭一, 荒井幸代 : 人流データに基づく避難者の適応戦略抽出, 第 31 回人工知能学会全国大会, 2M4-OS-32a-5in1 (2017.5 愛知)
9. (ポスター・査読なし) 浪越圭一, 荒井幸代 : 追従エージェントを考慮した人流データからの戦略抽出, Joint Agent Workshops & Symposium 2017 (2017.9 千葉)
10. (ポスター・査読なし) 浪越圭一, 荒井幸代 : 相対エントロピーに基づくモデルフリーマルチエージェント逆強化学習, Joint Agent Workshops & Symposium 2018 (2018.9 広島)
11. (ポスター・査読なし) 齋木匠, 浪越圭一 : マルチエージェントシミュレーションによるバイクシェアリングシステム導入効果の定量的評価, 第 19 回 MAS コンペティション (2019.3 東京)
12. (口頭発表・査読あり) 浪越圭一, 荒井幸代 : Multi-agent maximum discounted causal entropy 逆強化学習による報酬推定, 第 33 回人工知能学会全国大会, 3P4-J-7-05 (2019.6 新潟)
13. (ポスター・査読なし) 浪越圭一, 野田五十樹, 荒井幸代 : MAS モデル構築のための Heterogeneous swarm 逆強化学習の検討, Joint Agent Workshops & Symposium 2019 (2019.9 大分)

【受賞歴】

14. 歩行軌跡に基づく歩行者の行動規範の同定, ポスター発表優秀賞, Joint Agent Workshops & Symposium 2015 (2015.9)
15. 卒業論文「群衆の振舞データに基づいた個々の行動規範の抽出 ～災害発生時の避難モデルの生成～」, 優秀論文賞, 千葉大学工学部都市環境システム学科 (2016.3)
16. 追従エージェントを考慮した人流データからの戦略抽出, 優秀ポスター発表賞, Joint Agent Workshops & Symposium 2017 (2017.9)
17. 修士論文「群衆の行動ログを用いた行動戦略の推定 ～進化計算によるアプローチ～」, 工学研究科長賞・奨励賞, 千葉大学大学院工学研究科 (2018.3)
18. マルチエージェントシミュレーションによるバイクシェアリングシステム導入効果の定量的評価, 第 19 回 MAS コンペティション, ポスター発表セッション優秀賞受賞, 研究奨励金 5 万円 (2019.3)
19. MAS モデル構築のための Heterogeneous swarm 逆強化学習の検討, 最優秀ポスター発表賞, Joint Agent Workshops & Symposium 2019 (2019.9)

【研究資金・助成金獲得歴】

20. 平成 29 年度長谷川財団奨学金, 月額 3 万円 (2017.4～2018.3)
21. 平成 31 年度日本学術振興会特別研究員-DC2, 研究奨励費月額 20 万円・研究費 210 万円, 独立行政法人日本学術振興会 (2019.4～)

【その他】

22. 平成 29 年度特に優れた業績による返還免除, 半額免除, 独立行政法人日本学生支援機構 (2018.5)
23. 産業技術総合研究所人工知能研究センター社会知能研究チーム　技術研修員 (2017.7-2018.3, 2018.6～2019.3, 2019.8～2021.3 予定)
24. 産業技術総合研究所人工知能研究センター社会知能研究チーム　リサーチアシスタント (2017.7～2018.3, 2019.8～2021.3 予定)
25. 理研データ同化合宿（基礎編）修了（旅費･滞在費支援）, 理化学研究所計算科学研究機構 (2017.12 神戸)
26. 平成 30 年度リサーチ・アシスタント業務「大規模災害時における都市機能の適応的最適化人間の協調行動のインセンティブ設計」(2018.7-2019.3)
27. 千葉大学大学院工学研究院非常勤職員, 技術補佐員 (2018.11～2019.3)
28. 強化学習アーキテクチャ勉強会, 運営・幹事団 (2019.3～)