# モデルに内在する問題に対する
# マルチエージェント強化学習の設計

市川 嘉裕

電気通信大学 大学院情報理工学研究科

博士 (工学) の学位申請論文

2014 年 3 月

# モデルに内在する問題に対する<br>マルチエージェント強化学習の設計

博士論文審査委員会

主査　髙玉　圭樹　教授
委員　西野　哲朗　教授
委員　髙橋　治久　教授
委員　内海　彰　教授
委員　高橋　裕樹　准教授

# Abstract

This thesis focuses on the problems caused when unconsciously designing the model of the reinforcement learning agent such as Q-learning agent, and aims at exploring the methods that can solve such problems embedded in the model by investigating their effectiveness in the multi-agent environments. For this purpose, this thesis proposes the following three methods towards an acquisition of the global optimal policy by avoiding from acquiring the local minimum policies: (1) the learning progress arrangement method which promotes the agents to learn appropriately by turning the parameters of their learning speed (i.e., the learning rate and the discount rate) according to the difference of the learning progress among the agents; (2) the internal-reward-based method which enables the agents to acquire the higher reward than the current acquired reward by estimating the internal rewards from the external rewards ordinarily received from the given environment; and (3) the multiple policy archive method which stores the more than one policy (i.e., a set of the state-action pairs) that derives the Pareto rewards among the agents and utilizes the one of them to improve the performance. And, discussing of combining our all proposed methods and systematizing of problem embedded in models is the purpose of this paper. The intensive simulations of the simple multi-agent testbed problems have revealed the following implications: (1) a change of the discount rate parameter, gamma, contributes to avoiding the conflict situations caused by the interaction among the agents; (2) an estimation of the internal reward by subtracting the average reward from the external reward contributes to acquiring the higher reward than the current acquired reward by avoiding the local convergence of the learning; and (3) the archive of the multiple policies acquired in the process of searing the solutions contributes to exploring the global optimal policies by avoiding from acquiring the local minimum policies.

# 概要

本論文では，Q 学習に代表される強化学習手法をマルチエージェント環境に適用する際に，無意識に行っているモデル化に起因する問題に着目し，その解決手法の提案と有効性の検証を目的とする．特に，マルチエージェント強化学習では，他のエージェントの学習によって動的に環境が変化するため，エージェント間の複雑な相互作用を考慮したエージェントモデルの構築が必要であるが，それだけでなくエージェントモデルの前提（例えば，エージェントの学習速度は同じであるなど）として設定されていることによってエージェント間の協調に問題が生じ，大局的な最適政策の獲得が困難となることがある．このような問題は，システム設計者が予め自覚することは難しく，本論文ではこれを「**モデルに内在する問題**」と称し，エージェントの設計を改善することによってモデルに内在する問題（特に，マルチエージェント学習環境特有の同時学習問題や報酬の組み合わせが増えることで表面化する問題）の解決を図る．具体的には，(1) 一定の学習速度のモデルが原因で学習の停滞や獲得する政策（エージェントが学習した行動規則集合）の偏りを引き起こす問題，(2) 外部報酬に対する受容モデルが引き起こす局所的な政策獲得の問題，(3) 単一の政策のみによる学習モデルが望ましい政策の獲得を阻害する問題の解消に取り組む．また，いつ生じるかわからないモデルに内在する問題の性質上，様々な問題に同時に対処できることが望ましいため，上記の個別の解決策を統合することを試みる．さらに，モデルに内在する問題という観点からマルチエージェント学習を体系化し課題を整理することを試みる．

上記の目的達成に向け，(1) の問題に対しては，エージェントの行動がどれだけ確定的であるかを行動選択確率に関する情報エントロピーを用いて「**学習進度**」を定量化し，エージェント間で学習進度の差が大きくなり過ぎないように学習進度を共有しながら自身の学習速度を調整する手法を探究する．次に，(2) の問題に対しては，複数報酬問題において学習初期での獲得が容易で陥りやすい低い報酬への政策の獲得を避け，高い報酬を探索するために算出する「**内部報酬**」（外部報酬に置き換える目標）に基づいて状態-行動価値を見積もる（政策を学習する）手法を考案する．最後に，(3) の問題に対しては，学習途中で見つけた有望な政策（エージェント間の**パレート政策**）を複数保持し，それに基づいて低い報酬に対する学習を抑制することで，効率的に最適政策（パレート最適政策）を

探索する手法を提案する．

提案手法の有効性を検証するために，(1) 一定の学習速度のモデルに内在する問題を扱う例題（狭路すれ違い問題），(2) 外部報酬の受容モデルに内在する問題を扱う例題（マルチステップタスク割り当て問題），(3) 単一政策のみの学習モデルに内在する問題を扱う例題（マルチステップ 4 タスク問題）に提案手法を適用し，シミュレーション結果を通してその有効性を検証したところ，次の知見を得た：(1) 一定の学習速度のモデルに内在する問題に対しては，学習進度の違いがエージェント間の協調に影響を与えるが，通信を介して共有した学習進度を基に学習速度を調整する提案手法によって，エージェントの競合を回避できることを示した．特に，(i) 学習が進んでいるエージェントの割引率 $\gamma$ を下げる方法は，価値が高く選択されやすい行動価値を重点的に下げ，政策の偏りを防ぐことで，全てのエージェントが目標達成のために学習できる機会を増加させる働きがあること，(ii) 学習が遅れているエージェントの割引率 $\gamma$ を上げる方法は，目標達成につながる行動とつながらない行動の価値をはっきり分けるように推定するため，報酬獲得の機会が少ない状況から効率よく学習する働きがあることを明らかにした．次に，(2) 外部報酬の受容モデルに内在する問題に対しては，複数の報酬に対する報酬獲得の難易度の違いから局所的な政策に陥り易いが，外部報酬を基に見積もった内部報酬を用いて状態-行動価値を更新する提案手法によって，低い報酬へ向かう政策の獲得を避け，高い報酬へ向かう政策を獲得できることを示した．特に，(i) 高い報酬を集中的に探索するためには，今までに獲得した報酬の平均値を基準にして外部報酬を評価し直した内部報酬が有効であり，(ii) この内部報酬が最短経路の探索にも貢献することを見出した．最後に，(3) 単一政策の保持のモデルに内在する問題に対しては，多数の望ましくない報酬が望ましい報酬に対する学習を阻害するため，学習途中で見つけたパレート政策をアーカイブ保存し，それに基づいて状態-行動価値の更新を決定する提案手法によって，局所政策の獲得を回避できることを示した．特に，学習済みと判断したパレート政策のアーカイブを参照することによって，新しく学習する政策をそれよりも良い報酬を獲得できるものだけにすることを可能にしたことを示した．また，上記の三種の解決法を同時に例題に適用した実験の結果より，各手法の特徴が重なり合わさることによって特徴的な性能が得られることがあることを示した．さらに，モデルに内在する問題という観点からマルチエージェント学習を体系化することにより，今後の課題を明確にした．

# 目次

# 図目次

# 表目次

# 第 1 章

# 序論

## 1.1 マルチエージェント強化学習の意義とこれまでの研究

近年はエージェント（agent）が人間個人単位の生活を様々な側面から支援している．例えば，車載のカーナビゲーションや自動掃除ロボット，各種スマートフォンアプリなどがそれである．近い将来，我々の身近な範囲で賢いコンピュータプログラム同士が当たり前のようにコラボレーションし，さらにはそれらもが共存する図 1.1 のようなマルチエージェント社会が生じることが予想される．しかし，単体において賢いエージェントがコラボレーションしても必ずしもうまくいくわけではない．複数の主体が同一環境上で動作することで生じる様々な相互作用は，期待以上の良い効果をもたらす反面，競合のような全体として望ましくない事態が生じる可能性があるからである．マルチエージェントシステム（Multi-agent System: MAS）はそれぞれの行動規則に従う複数の自律的な主体（エージェント）が同じ環境上で動作するシステムの総称である．同じ環境上で複数のエージェントが動作するマルチエージェントシステムでは，エージェント間のミクロな相互作用が全体の秩序や挙動を形成し，反対に全体のマクロな秩序や挙動がエージェント間の相互作用に影響を与えるミクロ-マクロループが生じるとされている．このような現象によって，マルチエージェントシステムはシングルエージェントのシステムと比べて，係わるエージェントの数以上の成果をもたらすことが期待される．こうした全体に対して正の影響を及ぼすように動作するエージェントは互いに協力的・協調的な関係にあると見られ，これを意図的に引き起こすことは重要な研究対象とされる．一方で，マルチエージェントシステムは複雑な相互作用を前提とするため，工学的な利用を考えるシステム設計者が望むシステム全体の秩序や挙動（協力的・協調的な動作）を引き起こすように予めエージェントに適切な行動規則を組み込むことは困難である．その問題に対して，強化学習の導入によって適応的に行動規則（政策：policy）を獲得させる**マルチエージェント強化学習**が注目されており，盛んに研究されている [21][28][19][26]．強化学習 [20] では，エージェントは自身が観測した状態に対してなんらかの行動を実行し，変化した状況に応じて環境か


図 1.1　マルチエージェント社会のイメージ

ら与えられる報酬を基に，状態や行動の価値を見積もることで獲得する報酬を最大化するための政策を獲得することができる．元々未知の環境に対して適応する強化学習エージェントは未知の相互作用にも適応できることが期待される．実際，マルチエージェント強化学習では，学習が進み行動を変化させていく複数のエージェントの間で生じる相互作用の中でも，各エージェントは自身の報酬を最大化するような政策を獲得することに努める．しかし，シングルエージェントの環境を想定する手法として発展してきた強化学習をそのままマルチエージェント環境に適用する場合に生じる問題は少なくなく，そういったマルチエージェント強化学習の一般的な問題は，いくつかの課題としてまとめられている [2]．不完全知覚問題や報酬分配問題，同時学習問題などへの対処がその課題である．その一つである同時学習問題は，複数のエージェントが同じ環境上で動作し学習することから，状態遷移が他の学習エージェントという非定常的な要因により変化する環境となり，状態遷移の予測が難しくなるため，適切な行動の獲得が不可能な状況を生む可能性がある．この問題に対する先行研究として，全てのエージェントが報酬を共有するような純粋な協調問題してはマルチエージェント Q 学習はシングルエージェントの Q 学習（Q-learning）[22] と同様に学習が収束することが示されており [5]，一方では，他エージェントとの連帯行動を考慮する JALs（Joint Action Learners）[15] や，他エージェントの状態-行動価値を考慮し唯一つ存在するナッシュ均衡へ向けて学習するナッシュ Q 学習（Nash Q-learning）[10] など，学習の収束が示されているモデルが提案されている．しかし実応用を考えたときに対象となる問題は必ずしも上記のモデルが要求する仮定に当てはまるものばかりではない．例えば，各エージェントの報酬の最大化が全てのエージェントの報酬の最大化につながるという仮定やナッシュ均衡が唯一つである仮定，他のエージェントが実行した行動

が知覚できる仮定に従うことができない対象問題は多い.

## 1.2 本研究の着眼点：モデルに内在する問題

マルチエージェント強化学習はエージェントが個別に適応的に問題を解決する枠組みであることから，対象とする問題に対して環境やエージェントの設計が比較的自由である反面，効果的な解決を導く設計のためには解くべき問題の深い理解がシステム設計者に要求される．例えば，マルチエージェント強化学習を利用するシステム設計者は，エージェントが獲得報酬を最大化するために獲得する政策が，全体として望ましい振る舞いを形成するように環境とエージェントを設計する必要がある．しかし，各エージェントが独自に学習し，それらが複雑な相互作用を生むマルチエージェント環境では，システム設計者が設計の段階で問題の発現を自覚することは極めて困難である．これは強化学習ではエージェントが基本的に定常的な環境に対して学習する一方で，マルチエージェント強化学習では，他のエージェントの学習によって動的に変化する環境に対して学習するため，その学習過程が非常に複雑になるからである．そのような経緯で問題を自覚できない設計者によって暗黙的に設計された環境上で，正直に学習するエージェントは適切でない政策を獲得する．本研究では上記のように設計者が無自覚に不適切なモデルを設計してしまうことに焦点を当て，これによって生まれる問題を「**モデルに内在する問題**」と呼び，学習過程に生じる学習に対して悪影響となるエージェントのモデルの改良について考える．モデルに内在する問題の要因となる学習中に生じる問題に対して，学習中にだけ影響するような改良によって対処可能ならば，その改良が施されたマルチエージェント強化学習は対象となる問題に多くの仮定を要求することがなくなる．ここでは特に，従来研究の多くで無意識にあたりまえの設計として利用されるエージェントのモデルには改良の余地があると考える．例えば，最終的に運用されるエージェントの能力（入力状態や行動）が均一であるからといって，学習能力が均一である必要はないし，エージェントの学習によって変化する環境では設定された報酬に必ずしも従う必要はないし，運用される行動規則が一つであるからと言って学習中もそうである必要はない.

## 1.3 経路設計の例から見るモデルに内在する問題

ここでカーナビゲーションの経路設計の例を考える．各車に積まれたカーナビは独自に経路の混雑度などの状況を把握し，最適な（例えば最も空いている）経路を導き出す．ここで，仮に全てのカーナビが同じ経路を導き出すと，最適であったはずの経路がそうでなくなる（逆に混雑してしまう）ことが生じるため，大多数の車にカーナビを搭載するとむしろ混雑を招くことが報告されている [27]．マルチエージェント学習の観点で見ると，一方のエージェントが先に経路を決めてしまうといったように学習の進み具合に差が生まれ

ると，他方の学習が遅れたエージェントは先に経路を決めたエージェントに従う形で経路を導かなければならず，様々な経路の組み合わせが考慮されないことから，結果として一部のエージェントだけが最適とは程遠い経路しか導き出せない状況に陥る可能性がある．また，確率的に見出した一見して良さそうな経路に一度決めてしまうと，その他のより良い経路があったとしても探索されない状況を引き起こす．また，全てのエージェントが導いた経路がそれぞれにとって最良となることは現実的には少ないため，エージェント間のパレート解のようなお互いに妥協することで全体として良い経路（全体の混雑が低くなるような各自の設計経路）を見つけることができる必要がある．複数のエージェントが同時に学習する場合，上記の問題が常に生じる可能性があるため，これを防ぐために協調しながら学習することが重要になる．上に挙げた問題は一般的なモデルに内在する問題と捉え直すことができる．(1) エージェントの学習速度が一定であるというモデルは，学習が進み過ぎたり遅れ過ぎたりするエージェントが生じることが原因となって，他のエージェントの行動に依存した偏った政策しか獲得できなくなることや，場合によっては報酬を得ることさえできずに学習が停滞するような状況（要件を満たす行動がなく学習の指標になるものが得られず学習が進められない状況）を導く可能性がある．あるいは，(2) 報酬を与えられたままの大きさで学習に利用する報酬受容のモデルは，報酬と目標状態が複数存在するような状況で容易に局所的な政策（特定の報酬や目標状態だけに適応した政策）を獲得させる可能性がある．また，(3) 単一の政策のみの学習を扱うモデルは，全体にとって望ましくない（望ましいかどうかはエージェント視点では無自覚である）政策を獲得する方向に対しても一様に学習するため，結果として望ましい政策の獲得を阻害する可能性がある．

## 1.4 本研究の目的

本研究は，マルチエージェント強化学習におけるモデルに内在する問題を解決するエージェントの構築を目的とする．具体的には，上記に示した問題を打破する方法として，(1) 一定の学習速度のモデルが引き起こす学習する政策の偏りあるいは学習の停滞の問題に対して，エージェント間の学習の進行度合いの差を考慮して学習率と割引率を調整して学習を促進させるエージェント，(2) 複数の報酬に対する受容のモデルが引き起こす局所的な政策獲得の問題に対して，報酬関数から与えられる報酬（外部報酬）を直接使うのではなく，それまでの獲得報酬の記憶から内部的に変換した報酬（内部報酬）によってより高い報酬を探索するエージェント，(3) 単一の政策のみの学習のモデルが望ましい政策の獲得を阻害する問題に対して，エージェント間でのパレート報酬を考慮することによって複数の有望な政策を保持し，有効である行動系列とそうでない行動系列を区別して効率的に学習を進めるエージェントを探究する．また，いつ生じるかわからないモデルに内在する問

題の性質上，様々な問題に同時に対処できることが望ましいため，上記の個別の解決策を統合することを試みる．さらに，モデルに内在する問題という観点からマルチエージェント学習を体系化し課題を整理することを試みる．

## 1.5 論文の構成

本論文は以下のように構成される．

第 2 章において，強化学習の代表例として Q 学習を紹介し，マルチエージェント環境への適用とその問題点を整理し，本研究の位置づけを明確にする．

第 3 章において，モデルに内在する問題の詳しい紹介と，その問題を内包する具体的な例題のうち，本論文が扱う三種類の問題について詳述する．具体的には，(1) 一定の学習速度のモデルに内在する問題を扱う例題として，学習の進行度合いの差が大きくなるとデッドロックが生じる狭路すれ違い問題を，(2) 報酬値の直接の需要のモデルに内在する問題を扱う例題として，複数の報酬に対して確率的な獲得難易度の違いから局所的な政策に陥り易いマルチステップタスク割り当て問題を，(3) 単一政策のみの学習のモデルに内在する問題を扱う例題として，有効でない報酬の組み合わせが多数存在するマルチステップ多数報酬問題を導出する．

第 4 章において，一つ目のモデルに内在する問題に対処するため，エージェントの行動がどれだけ確定的であるかを評価するために状態における行動選択確率の情報エントロピーを用いて「学習進度」を定量化し，エージェント間で学習進度の差が大きくなり過ぎないように学習進度を共有しながら自身の学習速度を調整するエージェントを提案し，例題に適用する実験によってその有効性を検証する．

第 5 章において，二つ目のモデルに内在する問題に対処するために，複数報酬問題において低い報酬への収束を避け，高い報酬を探索するための三つの方法により算出される内部報酬に基づいて行動-状態価値を見積もるエージェントを提案し，例題に適用する実験によってその有効性を検証する．

第 6 章において，三つ目のモデルに内在する問題に対処するために，行動選択確率が確定的になった政策をアーカイブへ保存し，その後それを利用することで有望でない政策の学習を抑制することで大局的な最適政策の獲得を目指すエージェントを提案し，例題に適用した実験によってその有効性を検証する．

第 7 章において，モデルに内在する問題への個別の対処方法の単純な統合が複合的なモデルに内在する問題を解決できるかどうかという疑問に対して，ここまでの三つの提案手法を実際に統合し，例題へ適用することでその調査を行う．

第 8 章において，モデルに内在する問題の体系化を試みる．具体的には，従来のマルチエージェント学習の問題をマルチエージェント学習モデルの設計の三段階に対応するよう

に分類して捉えなおすことにより，モデルに内在する問題の実態を明らかにする．

第 9 章において，上記の各章で得られた知見をまとめ，本研究の成果について述べるとともに，今後の課題について述べる．具体的に，(1) モデルに内在する問題の体系化で示した分類のモデルを考慮した総合的なマルチエージェント学習モデルの設計論の構築，(2) 個別対処された手法の統合のための方法論の検討などについて述べる．

# 第 2 章

# マルチエージェント学習

強化学習（reinforcement learning）[20] では，エージェントは自身が観測した状態に対してなんらかの行動を実行し，変化した状況に応じて環境から与えられる報酬を基に，状態や行動の価値を見積もることで獲得する報酬を最大化するための政策を獲得することができる．マルチエージェント強化学習では，学習が進み行動を変化させていく複数のエージェントの間で生じる相互作用の中でも，各エージェントは自身の報酬を最大化するような政策を獲得することができる．本章では，強化学習を概説し，その後にマルチエージェント環境への適用とその問題点を述べ，最後に本研究の位置づけを明確にする．

## 2.1 強化学習

### 2.1.1 強化学習におけるモデル化

強化学習では通常，マルコフ決定過程（MDPs : Markov Decision Processes）によって以下のようにモデル化された問題が扱われる．

- $\mathcal{S}$（状態集合）
- $\mathcal{A}$（行動集合）
- $\mathcal{P}$（状態遷移確率）あるいは $\mathcal{T}$（状態遷移関数）
- $\mathcal{R}$（報酬関数）

後述するマルチエージェントの強化学習と比較するために，シングルエージェントの強化学習のモデルを図 2.1 のように示す．図に示すように，強化学習においてはエージェント（agent）のモデルと環境（environment）のモデルの二つに大別できる．ここで，エージェントは状態 $s \in \mathcal{S}$ を知覚し，行動 $a \in \mathcal{A}_s$ を出力することで，状態遷移規則 $\mathcal{P}$（あるいは $\mathcal{T}$）に基づいて状態が遷移し，報酬関数 $\mathcal{R}$ として定義されている状態に応じた報酬 $r \in \mathcal{R}$ が与えられる．エージェントは獲得した報酬によって状態価値や状態-行動価値を

更新し，最適な（単位時間当たりの獲得報酬を最大化する）政策 $\pi$（policy）を学習することを目的とする．政策は一般的に状態価値や状態-行動価値に基づいて算出される各状態における各行動の選択確率であり，強化学習においては価値の更新とそれに基づく行動選択の繰り返しによる試行錯誤を通して政策を学習する．MDPs の枠組みの中では，エージェントは全ての状態を間違いなく認識できるため，環境の状態とエージェントが知覚する状態は完全に一致する．このような状況でエージェントと環境を区別することに大きなメリットはないが，区別することによってエージェントが環境上に位置する存在であると捉えれば，例えば知覚情報が制限されたエージェントなどを考慮する枠組みへの拡張が容易に可能であり，他の学習エージェントが別に存在するマルチエージェントの枠組みもごく自然に拡張可能であると言える．

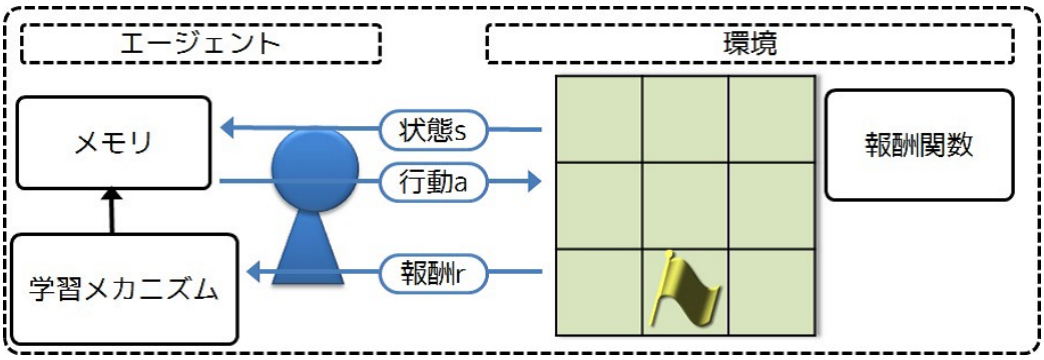



図 2.1　強化学習のモデルにおけるエージェントと環境間の相互作用

### 2.1.2　基本となるエージェント

本論文では，強化学習エージェントとして代表的な強化学習法である Q 学習（Q-learning） [22] によって学習するエージェントを扱う．ここでは基本となる Q 学習エージェントの具体的な設計について記述する．図 2.2 に示すように，エージェントは観測しうる状態 $s \in \mathcal{S}$ ととり得る行動 $a \in \mathcal{A}$ の任意の組 $(s, a)$ に関する状態-行動価値 $Q(s, a)$ を見積もることで獲得する報酬を最大化することが目的である．そのためにエージェントは環境と呼ばれる状態空間において自身の現在の状態（$s : state$）を観測し，なんらかの行動（$a : action$）を出力し，環境からの報酬（$r : reward$）を基に $Q(s, a)$ の更新を繰り返すことで学習を進める．$Q(s, a)$ は，状態の観測と行動の出力という一連の動作の後に環境から与えられる報酬を基に逐次的に次式を用いて更新される．

$$Q(s,a) \leftarrow Q(s,a) + \alpha \left[ r + \gamma \max_{a' \in \mathcal{A}'} Q(s',a') - Q(s,a) \right] \tag{2.1}$$

この式において，$s$ は状態，$a$ は行動，$r$ は報酬，$s'$ は次状態，$a'$ は次状態における行動，$\mathcal{A}'$ は次状態においてとり得る全ての行動の集合，$\alpha$，$\gamma$ は Q 学習の学習パラメー

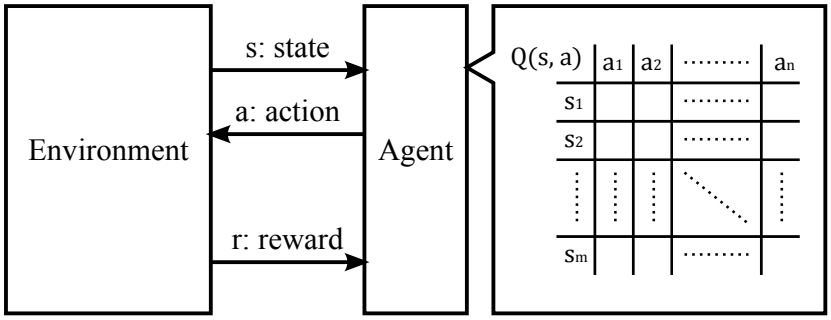



図 2.2　Q 学習エージェントのアーキテクチャ

タ $[0 \leq \alpha, \gamma \leq 1]$ であり，それぞれ学習率（learning rate）と割引率（discount rate）と呼ばれる．また，$\max_{a' \in \mathcal{A}'} Q(s', a')$ は状態 $s'$ において価値が最大となる行動 $a'$ の状態-行動価値であり，これにより $Q(s, a)$ がすぐに得られる報酬だけでなく，将来的に得られる報酬を含めた期待報酬値であることを意味している．学習率 $\alpha$ はそれまで学習した状態-行動価値（$Q(s, a)$）に対して今回得られた報酬および将来の報酬を含めた期待報酬値（$r + \gamma \max_{a' \in \mathcal{A}'} Q(s', a')$）をどれだけ重くみるかの割合を表し，割引率 $\gamma$ は将来的に得られる報酬をどれだけ重くみるか，実際はどれだけ割引くかの割合を表している．

状態-行動価値に基づく政策 $\pi$ は，試行錯誤的な学習のなかで，価値を徐々に推定しながら多くの報酬を得ることのできる行動を優先的に選択することで，できるだけベストに近い行動をとりながら効率よく環境に適応するような学習を生み出す．全ての行動が選択される可能性があれば，無限時間の学習と学習率の調整により最適な状態-行動価値が獲得できると示されている [22].

どの問題においても二次元離散座標の環境上を移動しながら目標座標へ到達することを目的とする本論文のエージェントはさらに以下の設計から成る．状態集合 $\mathcal{S}$ は，自分と他の全てのエージェントが存在する座標を完全知覚し，それらのとり得る全ての組み合わせとする．また，行動集合 $\mathcal{A}$ は，隣り合う座標に移動するかしないかの全ての選択肢である｛上移動，下移動，右移動，左移動，移動しない｝とする．さらに，エージェント同士が同じ座標に重なることはできないことに加えて，エージェントが座標の外に出ることもできないこととし，そのような行動を選んだエージェントは動かずにその場に留まる．また，目標座標（ゴール報酬の得られる位置）に到達したエージェントはそのエピソードの間はその場に留まり続け，行動選択や学習を行わない．そのため，二体のエージェントは同じ目標座標を候補としてもっていても同時その目標に到達することはない．１エピソードは初期状態から報酬を獲得するまで（正確には，終状態へ到達するまで）の期間を表すが，エージェントが持つ政策によってはいつまでも報酬を獲得できないことが起こる．この対処として，１エピソードには最大ステップ数という定数を用意し，一定のステップ数が経過しても報酬が得られない場合は強制的にそのエピソードを打ち切り，次のエピソー

ドに移行するものとする．ステップは行動を一回実行する時間の単位を表す．

行動選択手法は式 (2.2) に示す状態-行動価値を基にしたボルツマン分布に基づく選択（以下，ボルツマン選択）を採用する．$\pi(s,a)$ は状態 $s$ において行動 $a$ が選択される確率を表す．

$$\pi(s,a) = \frac{\exp(Q(s,a)/T)}{\sum_{b\in\mathcal{A}} \exp(Q(s,b)/T)} \tag{2.2}$$

$T$ は温度パラメータと呼ばれる定数であり，$T$ が高いほど行動選択のランダム性が高く，0 に近いほどランダム性は低くなる．本論文で扱うエージェントは，Arai ら [1] が用いた次式により定められる $T$ を採用する．

$$T = 0.5 \times 0.998^{episode} + 0.01 \tag{2.3}$$

$T$ はエピソードの関数になっており，エピソードが小さく，温度 $T$ が高い学習初期においては $Q(s,a)$ の大きさに関わらずほぼランダムに行動が選択される一方，エピソードの増加に伴い温度 $T$ が低くなると，$Q(s,a)$ が相対的に大きい行動が選ばれやすくなる．このようなエピソードに応じて減少する $T$ は，学習初期の状態の探索を保証し，徐々にランダムを低くすることで学習下政策を反映した行動を実施するための設定であり，一般的に多くの研究でも用いられている．

## 2.2 マルチエージェント強化学習

### 2.2.1 マルチエージェント強化学習におけるモデル化

マルチエージェント強化学習とは，複数の強化学習エージェントが同じ環境上で独立に学習し，個々に獲得報酬の最大化することで，全体の最適な振る舞い（対象とする問題を解決するための個々の政策）を獲得する枠組みである．マルチエージェントシステムはシングルエージェントのシステムと比べて，係わるエージェントの数の増加以上の成果をもたらすことが期待される一方で，マルチエージェントシステムは複雑な相互作用を前提とするため，工学的な利用を考えるシステム設計者が望むシステム全体の秩序や挙動（協力的・協調的な動作）を引き起こすように予めエージェントに適切な行動規則を組み込むことは困難である．それに対してマルチエージェント強化学習では，学習が進み行動を変化させていく複数のエージェントの間で生じる相互作用の中でも，各エージェントは自身の報酬を最大化するような政策を獲得することができる．この利点は，人間が解くべき問題の全容を把握することなく，部分的にモジュール（エージェント）単位に目的を定めることで，エージェントが自律的に問題を解決することに結びつく点である．

図 2.3 にマルチエージェント強化学習のモデルを示す．図は青のエージェントの観点でのモデルを表している．図中で赤のエージェントが環境の枠内に置かれているように，自

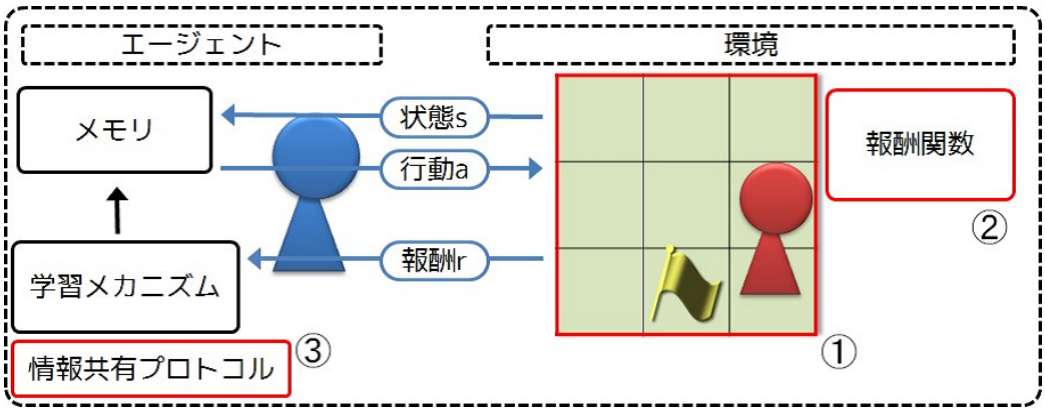



図 2.3　マルチエージェント強化学習のモデルにおけるエージェントと環境間の相互作用

分以外のエージェントは環境の一部として捉えることができる．マルチエージェント強化学習がシングルエージェントの強化学習と異なる点は，(1) 状態遷移が自身の行動だけに依らず他のエージェントの行動に依存すること，(2) 自身の状態だけでなく他のエージェントの状態に依存して報酬が与えられることが多くあること，(3) 情報共有プロトコルを利用した他のエージェントとの明示的な協力関係を考慮できることである．このように，マルチエージェント強化学習では前述のシングルエージェントの強化学習で考慮すべき事項に加えて他のエージェントとの関係を考慮する必要が生じる．言い換えれば，複雑なインタラクションの生じる環境下で最適な政策を獲得するためにシステム設計者が注意深く考慮しなければならない設計項目が格段に増加する．例えば，学習速度の設定，報酬の設計，保持する情報の設計など，これらのどれを取っても一体のエージェントにとって適切な設計が決まっても他のエージェントには不都合な設計となる可能性は少なくない．具体例として，エージェント全体の報酬最大化をチームによるプロジェクト成果の最大化に置き換えて説明を試みる．この例によって，同じ対象問題に対する複数のモデル化によるアプローチとそれらがもつ問題を示す．例えば，(1) チームが成果を上げた（全体として一つの報酬が得られた）とき，誰のどの行動が成果に貢献したかは正確に知ることはできないため，誰がどれだけの報酬を与えられることが適切かわからず，結果として働き者とそうでない者が生まれる可能性がある．一方で，(2) 個人の成果が（役割分担と達成度合いというような形で）定量化できれば個人の成果の合計を全体の成果と考えるこができるが，この場合，誰にどの役割を担当させることが適切かどうかの保障が必要であり，さもなければ，全員が成果を最大に上げてもそれが最適かどうかは保障されない．さらに別の方法として，(3) 役割分担までも流動的に決めるモデル化を考えることができる．この場合，上手くいけば最適な役割分担を獲得することができるが，ある時点で上がった成果（報酬）をもとに流動的に形成された役割分担が最適である保証はない．例えば，個々人の仕事への慣れの早さ（学習速度の設計）や仕事ごと難易度の差異（目標状態の設計）の

影響でたまたま落ち着いた（局所的に良いだけの）分担であるかもしれないからである．あるいは，質の違う数種の仕事が成果という一次元で評価されることによって，特化した一つの局所的な分担から抜け出せずにより良い組み合わせが考慮されないようなケースも考えられる．上記の (1) と (2) のモデル化で生じる問題は一般に報酬分配問題と呼ばれて分類されており，(3) は同時学習問題と呼ばれて分類されている．これにさらにもう一つの問題として不完全知覚問題と呼ばれるものを加えて，マルチエージェント強化学習の問題を次節でまとめる．

### 2.2.2 マルチエージェント強化学習の問題

荒井 [2] はマルチエージェント強化学習の課題として特に次の三つの問題を取り上げている．

- 同時学習問題
  この問題は一つの環境内で複数の学習エージェントが同時に存在することによって引き起こされる問題である．状態遷移が他の学習エージェントという非定常的な要因により変化する環境では状態遷移の予測は難しく，学習が著しく阻害され，場合によっては適切な行動の獲得が不可能な状況を生むため深刻な問題である．図 2.4 に示すハンター問題 [4]（獲物を追い詰める複数エージェントのタスク）の例では，息が合って左右から挟み込むように獲物を捕らえることを学習できることもあれば，学習の進め方によっては息が合わずなかなか獲物を捕らえる行動の学習がされない可能性がある．例えば，自分が下から追い詰めても上から他のエージェントが来なければ捉えることができないため，方針を変えて左から追い詰めようとしたところ，他のエージェントの方も同じ考えで方針を同時に変更するように，同時に学習していることによって効率的でない行動を取る状況が生じる．また，学習の進められ方によってはたまたま上手くいった行動に対して局所的な政策の学習を進めることで他の政策が探索されなくなる可能性がある点でも重要な問題である．
- 報酬分配問題
  この問題はエージェント個別の行動評価（報酬分配）の困難さによって引き起こされる問題である．複数のエージェントの行動の結果得られた報酬を個々のエージェントの行動に対して適切に与えることは難しく，例えば全ての行動は報酬に間接的に関与していると考えればその難しさは明白であり，報酬が適切に与えられなければ適切な行動の獲得が保証できないことに繋がるため深刻な問題である．図 2.5 に示すハンター問題の例では，まず赤のエージェントが獲物に接近して獲物の動きを制限し，青のエージェントが獲物を挟み込むことで捕えることに成功した状況を表しているが，この時初めに獲物の動きを制限した赤を高く評価するべきか，最終的

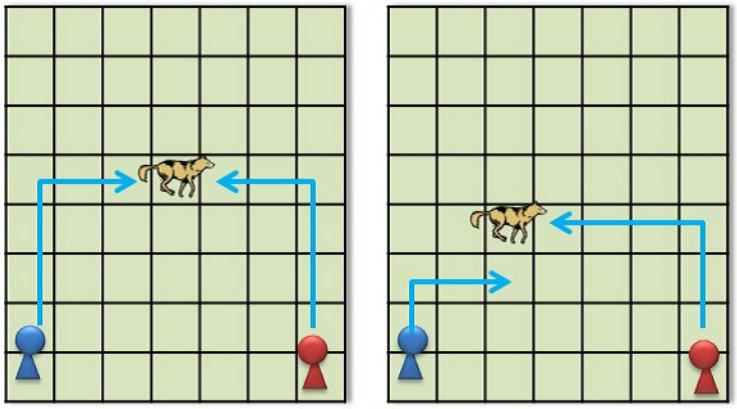


図 2.4　ハンター問題における同時学習問題

に捕えられたのは青の行動なくしては不可能なのでどちらも均等に評価するのかという異なる評価を考えることができる．獲物に先についたエージェントを高く評価することで全体として早く獲物を捕らえる行動を学習できる可能性があるなど，報酬の分配の仕方によって最終的に異なる学習がされることになる点でも重要な問題である．

- 不完全知覚問題
  この問題はエージェントの不完全な知覚がもたらす誤った状態遷移の認識によって引き起こされる問題である．一般に大規模な状態空間を対象とするマルチエージェント環境では全ての状態を別々の正しい状態に知覚できないことが不完全知覚問題を引き起こす．図 2.6 に示すハンター問題の例では，周囲 25 マスを知覚するエージェントが獲物のいる方向に対して間違って認識される問題が起こる．例えば，左の状況と右の状況は赤と青のエージェントにとってどちらも同じ状態であると知覚されるため，左側の状況を経験したあとでは，上に獲物がいると学習してしまい，結果として右側の状況下でも上へ移動する行動を選んでしまう．

マルチエージェント強化学習を利用するシステム設計者は，エージェントが獲得報酬を最大化するために獲得する政策が，全体として望ましい振る舞いを形成するように環境とエージェントを設計する必要がある．マルチエージェント強化学習はエージェントが個別に適応的に問題を解決する枠組みであることから，対象とする問題に対して環境やエージェントの設計が比較的自由である反面，効果的な解決を導く設計のためには解くべき問題の深い理解がシステム設計者に要求される．しかしながら，各エージェントが独自に学習するマルチエージェント環境では，システム設計者が設計の段階で問題の発現を自覚することは極めて困難である．これは強化学習ではエージェントが基本的に定常的な環境に対して学習する一方で，マルチエージェント強化学習では，他のエージェントの学習に

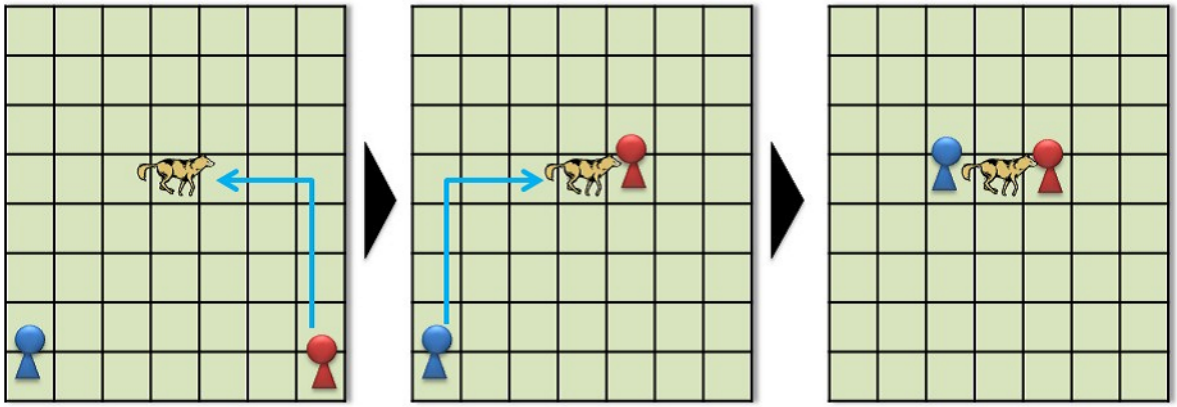


図 2.5　ハンター問題における報酬分配問題

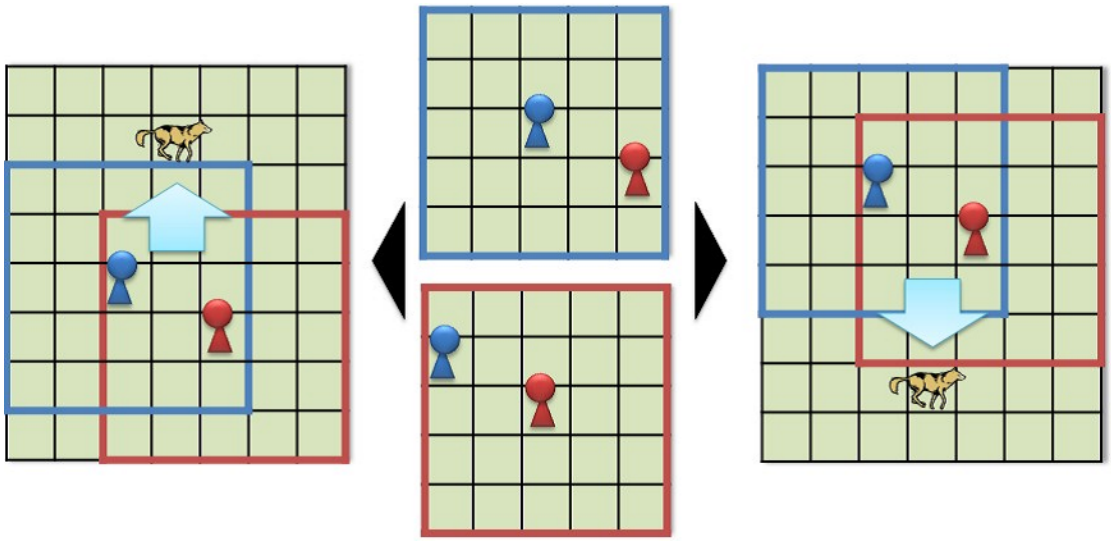


図 2.6　ハンター問題における不完全知覚問題

よって動的に変化する環境に対して学習するため，その学習過程が非常に複雑になるからである．そのような経緯で問題を自覚できない設計者によって設計された環境上で，正直に学習するエージェントは適切でない政策を獲得する．

実応用の観点からみると，マルチエージェント学習の構成要素を一から設計することは稀であり，部分的な設計が与えられている場合がほとんどである．例えば，制約条件としてのエージェントの出力に対する制約や状態数やその遷移確率などは問題によって与えられる場合が多い．その他，エージェントの数も例えばサッカーの問題であればメンバーの数は決まっているし，ロボットアームの最適化の問題では関節の数によりそれが決定する．それらに対して，エージェントの学習速度や報酬などは比較的自由に設計することが可能である．例えば，最終的に運用されるエージェントの能力（入力状態や行動）が均一であるからといって，学習能力が均一である必要はないし，エージェントの学習によって

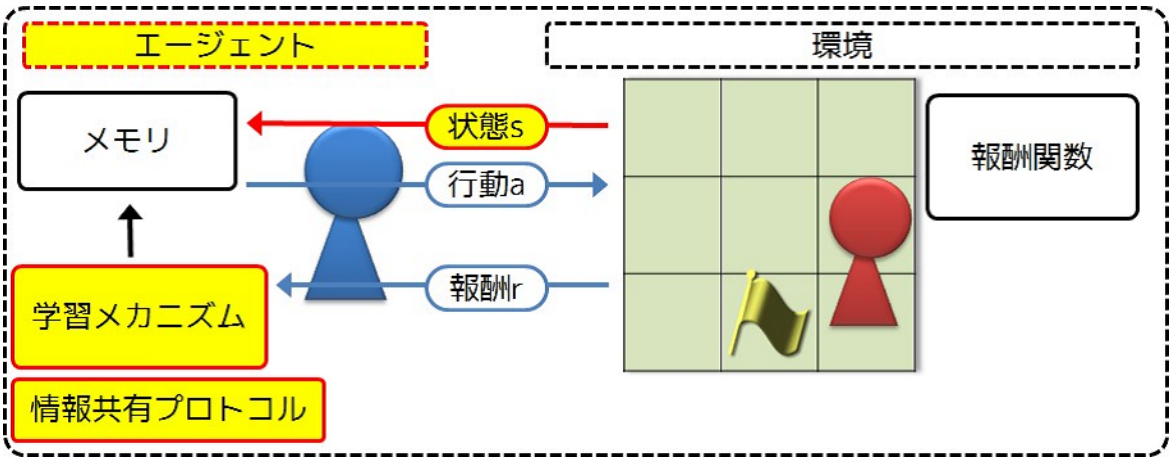



図 2.7　従来研究が主に対象としてきた領域

変化する環境では設定された報酬に必ずしも従う必要はないし，運用される政策が一つであるからと言って学習中もそうである必要はない．これらに共通した問題として挙げられるのは学習が完了した状態に至るまでの過程を考慮せずに設計されることであるが，上でも述べたように複雑なインタラクションを通して進められる学習過程を考慮して設計することは実質的に困難である．こういった設計の時点で考慮しにくいため対処しきれず，やってみるまで分からないような無意識的に内在する問題はこれまであたりまえな問題として直接的に対処されてこなかった．具体的に，対処されてきたのはエージェントの観点からみれば，図 2.7 で黄色で示す部分がほとんどである．例えば Tan[21] は情報共有プロトコルとして知覚状態や政策の共有をするエージェントの分析を行った．

## 2.3　本研究の位置づけ

マルチエージェント強化学習では，他のエージェントの学習によって動的に変化する環境に対して学習するために非常に複雑な学習過程が生じることから，それによって引き起こされる問題をシステム設計者が予め自覚することは難しく，そのようにして設計されたモデルにおいて正直に学習するエージェントは適切でない政策を獲得する．上記の問題を含むマルチエージェント学習で生じる問題は，究極的には対象問題に対してエージェントを用いて (1) 何を学習させたいのか？，(2) 何を頼りに学習させるのか？，(3) どのようなメカニズムをもって学習させるのか？が正しくモデル化できさえすれば解決される．このことから本研究ではマルチエージェント学習で生じる問題をモデルが引き起こす問題であると捉える．本研究は，この観点から捉えた諸々の問題を「**モデルに内在する問題**」と呼び，中でも特にエージェントが持つ学習メカニズムのモデルに内在する問題の解決を図る．すなわち状態空間や報酬関数が正しく設計された環境が与えられる問題に対して，

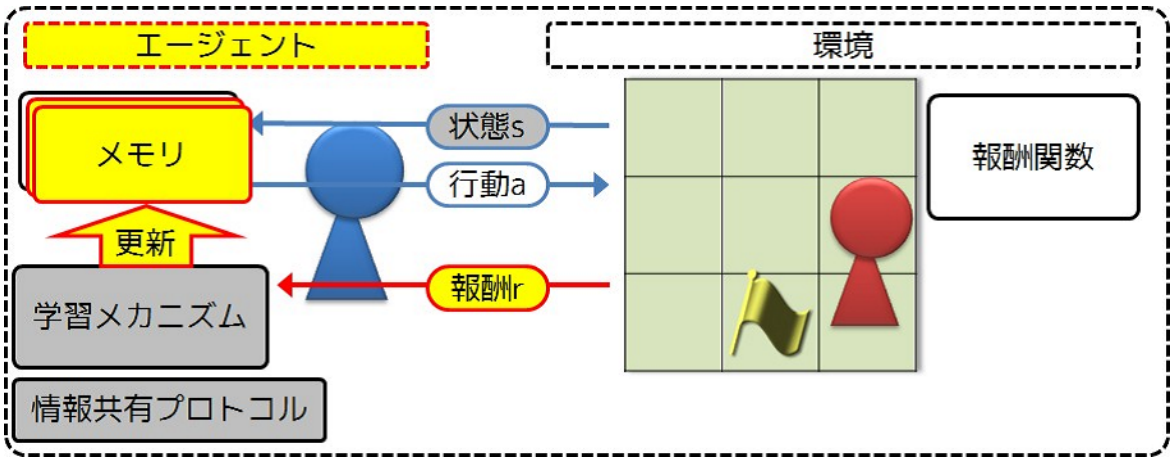



図 2.8　本研究が対象とする領域

エージェントが各々の獲得報酬の最大化を目指すことで全体の最適化を図るマルチエージェント強化学習を扱う．第一段階としてこれを扱うのは，マルチエージェント学習には必ず伴う同時学習問題に対応できる学習エージェントのモデルの構築が重要であるためである．状態空間や行動空間，報酬関数，環境の制約と比べて，学習メカニズムは対象問題への依存性が少ないため，このモデルの改良によって対処できれば，状態空間や報酬関数のモデルの改良と比べて対象問題要求する仮定が少ないと考えられる．学習メカニズムでは特に，従来研究の多くで無意識にあたりまえの設計として利用されるエージェントのモデルには改良の余地がある．なぜなら，学習メカニズムの多くはシングルエージェント環境で培われてきた技術を借用しているため，マルチエージェント環境初めて起こるようになった問題全てにはサポートされていないからである．本研究では従来から暗黙のモデルとして用いられている設計箇所（図 2.8 で黄色で示す部分）に対して，そこから生じる問題を解消するための改善を加えたエージェントの構築を目指す．これは，様々な種類の手法が開発されてきたエージェントモデルに共通的にあるにもかかわらず，ほとんど触れられてこなかった問題へ本研究が網羅的に取り組むことを意味する．具体的に対処するモデルとその問題点として次のようなものを挙げる．ただしここでは問題の概要だけに触れ，詳細は次章で述べる．

- 一定の学習速度のモデルに内在する問題
  環境中でエージェントに適した学習率や割引率の設定は何であるかという問題はシングルエージェント強化学習の中でも見られるが，マルチエージェント強化学習では他のエージェントとの学習の進み具合の関係によって他のエージェントの学習を阻害する可能性が高まる．学習が早すぎたり遅すぎたりするエージェントが混在することで報酬を得ることすらできないような状況に陥る可能性もある．これらはエージェントが一定の学習速度を持って学習していることに起因する．これは図

2.8 の「更新」の部分に相当する．更新は本来メカニズムの内の内容であるが中でも特に更新の部分に焦点を当てる．

- 報酬の受容のモデルに内在する問題
  学習エージェントは通常，単一の目標に対して効率良く学習ができるように設計されているため，目標状態が複数存在する複数報酬問題では容易に局所的な最大値をもつ政策に陥る可能性がある．これは環境から外部的に与えられる報酬値をそのままの形で利用することに起因する．これは図 2.8 の「報酬」を受けている部分に相当する．
- 単一政策の保持のモデルに内在する問題
  エージェントが複数の報酬に対して個別に報酬最大化を目指すとき，エージェントは様々な報酬を受け取った経験から学習した期待できる報酬値の高い行動を取りながら学習すると，低い報酬のある状態付近を探索しなくなる．そのような理由から，自身の報酬が低くても他方のエージェントの報酬が高いといったようにエージェントが得る報酬間にトレードオフ関係がある場合，全体として最適な政策の獲得が阻害される．また，そのような報酬の組がある場合は必ず複数の報酬の組が考えられ，それらを得るための政策の組も複数必要になるにもかかわらず，既存の学習法では単一の政策のみしか扱わないため，トレードオフ関係にある報酬群を同時に考慮して学習されていないことを意味する．これは図 2.8 の「メモリ」の部分に相当する．

# 第 3 章

# モデルに内在する問題

本章では，本研究で取り組むいくつかのモデルに内在する問題について詳しく述べ，それらが顕著に現れる例題を提示する．これに取り組み，検証を通してより本質的な問題の解決を図る．具体的には，第 2 章で述べた，マルチエージェント強化学習のモデルの全体からみると図 3.1 の黄色の部分に相当する問題を扱う．

(更新) 一定の学習速度のモデルに内在する問題
(報酬) 報酬の受容のモデルに内在する問題
(メモリ) 単一政策の保持のモデルに内在する問題

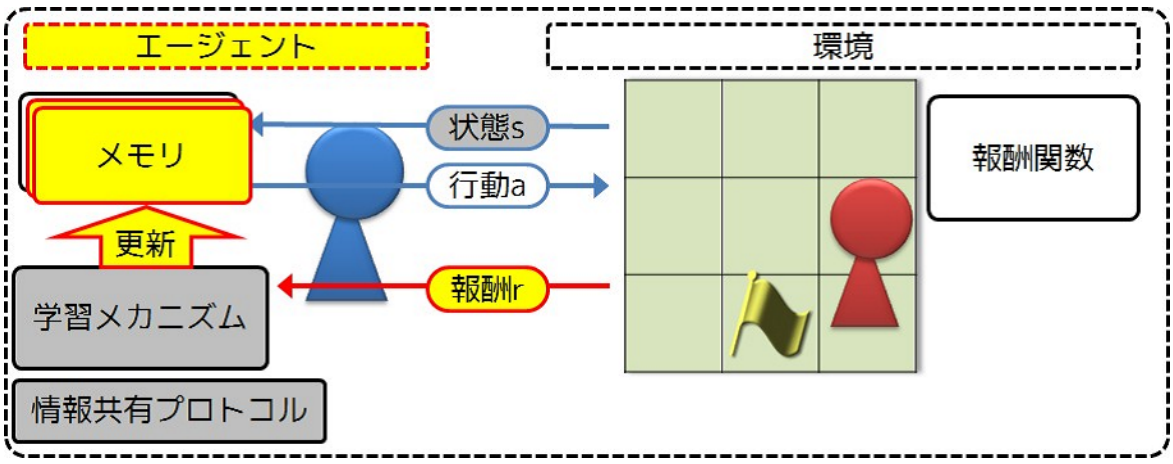



図 3.1　本研究が対象とする領域

## 3.1 一定の学習速度のモデルに内在する問題

### 3.1.1 問題

エージェントの学習が進むということは効率的に報酬を獲得することであり，そのためには確定的に行動選択することが必要である．確定的な行動選択は様々な状態の探索の幅を制限しているという見方もできる一方で，学習が上手くいかなくなった場合，探索の度合いが適切でないため調整される必要がある．例えば，ボルツマン選択において温度パレメータ $T$ の適切な変更がこれにあたる．一方で，マルチエージェント環境ではエージェント自身の学習進度のみに依らず他エージェントの学習進度によって状態遷移の制限が引き起こされる．あるいは全体の状態の組み合わせによって報酬が異なる環境の場合，獲得報酬を大きくするために学習の進んでいる相手の政策に追従するため，他エージェントの学習進度は間接的にも獲得する政策に影響する．以上のことは学習進度が同程度のエージェントからよりも相対的に学習が進んでいるエージェントに引き起こされる可能性が高い．例えば学習が進んでいない（行動が確定的でない）エージェントの方が状態遷移の確かさが低く，他エージェントの状態遷移を阻害する可能性が低い．学習進度に差が生じる問題は，エージェントの学習速度を定める学習パラメータや温度パラメータが同じに設定されれば解決するわけではない．エージェントの設計が全く同様であっても，例えば初期状態や目標状態への距離，行動順などの違いによってエージェントの学習過程は全く違ったものになるため，学習の進み方も異なるのである．そのようにして生まれる学習進度の差異が引き起こす問題は学習パラメータなどの予めの設定による解消は難しい．

図 3.2 は二体のエージェントが同じ旗の位置に到達することで両者に報酬が与えられる問題であるとする．設定として上のほうにある旗に到達して与えられる報酬の方が大きく，大局的な最適政策である問題とする．エージェン達ははじめどちらに対するゴールもいくらか経験し学習を進めるが，学習速度が均一であることを仮定すると，青のエージェントがスタート座標に近い下の旗へ向かう政策を先に強める．すると青のエージェントは下の旗へ向かうことが多くなり，それよりもどちらの旗へ向かうか決められていなかった赤のエージェントは，青のエージェントの影響で報酬が得られる可能性が高くなっている下の旗へ向かうように学習を進める．そのようにして下の旗へ向かう可能性が高まると学習が進められて連鎖的に下の旗へ向かう政策が強められる．これは大局的な最適政策とは異なる．この問題に対して，図 3.3 に示すように，赤のエージェントの学習の速さを早め，あるいは青のエージェントの学習の速さを遅くすることで，青のエージェントが下の旗へ向かう政策を強めるよりも早くに赤のエージェントが上の旗へ向かう政策を強められるようになれば，前述の学習結果とは逆に大局的な最適政策を獲得することになる．以上のように学習速度を適切な不均一な値に設定することができれば，局所政策の獲得を避け大局

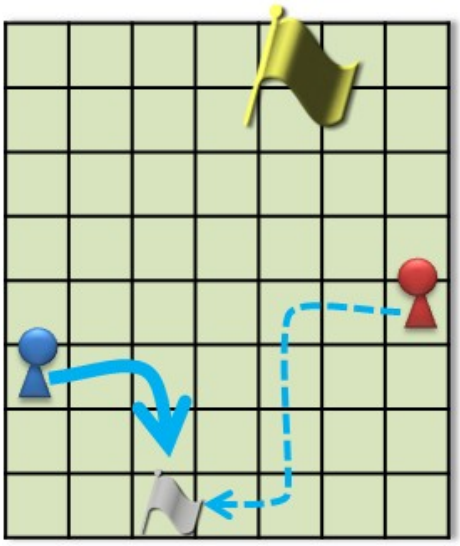


図 3.2　一定の学習速度のモデルに内在する問題における局所政策獲得

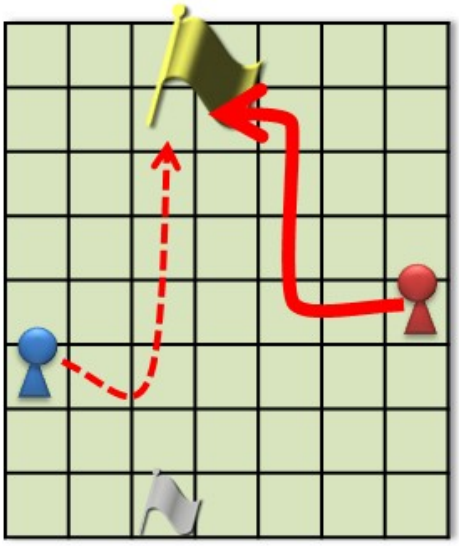


図 3.3　一定の学習速度のモデルに内在する問題の解消

的な最適政策を獲得できると考えられる．しかし，この問題を自覚することができるのは学習中であることから，予めある一定値に設定することは難しいため，学習中に調整するような対処が必要である．

### 3.1.2　例題

ここでは，一方のエージェントだけが学習を進めることができる状況が生じ，それが他エージェントの状態遷移を阻害することで結果として全体にとって最適な（全てのエー

ジェントが同時に自身の目標を達成できる）政策が獲得できない可能性があるものを例題として扱う．狭路すれ違い問題 [1] は，二次元離散座標の状態空間中で独自の目標状態（互いに相手に知らされない）への道のりが交差しており，かつ，互いに同じ格子に侵入できない制約があるため，どちらかが脇道に避けて相手が通過するのを待つ動き（すれ違い）を学習しなければならない複数エージェントのタスクである．

狭路すれ違い問題は図 3.4 のような二次元離散座標で表される．二体のエージェント A，B それぞれの初期座標は StartA，StartB，目標座標は GoalA，GoalB として表されている．この環境におけるエージェントの学習目標はそれぞれの初期座標から同時に出発して両方のエージェントが目標座標へ到達することである．

エージェントは 1 ステップ（step）と呼ぶ単位離散時間に一つの行動を実行する機会が与えられる．このとき 1 ステップにおいてエージェント A，エージェント B の順番に行動し，全てのエージェントの行動の結果，エージェント毎に別々の報酬が環境の応答として与えられる．報酬は二種類あり，一つ目はエージェントが目標座標へ到達したステップに与えられるゴール報酬で，二つ目はそれ以外のステップに与えられる通常報酬である．具体的には，ゴール報酬 $r_G = 5$，通常報酬 $r_N = -0.01$ とする．また全てのエージェントがゴール報酬を得ると，エージェントは初期座標に戻され，引き続き学習を行い定められた回数（例えば `MAX_EPISODE`= 500）だけ繰り返す．この繰り返し一回をエピソード（episode）と呼び，ゴール報酬は 1 エピソードの間では各エージェントに対して一回だけ，目標座標へ到達したステップにおいてのみ与えられる．また，初期座標に戻された状態から一定ステップ（例えば `MAX_STEP`= 100）経過しても全てのエージェントがゴール報酬を得ることができないとき，そのエピソードは強制的に打ち切られ，次のエピソードへ移行する．エピソードの開始時には必ずエージェントは初期座標に戻される．

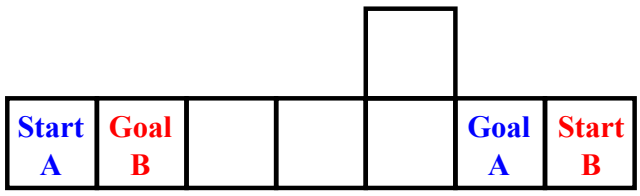



図 3.4　狭路すれ違い問題

### 3.1.3　強化学習エージェントの挙動と改善の要件

強化学習エージェントが上記の設定の下で学習を行った場合，初期座標を StartB とするエージェント B だけが目標状態へ辿り着き，エージェント A が目標状態へ辿り着けず学習も進められないという結果が得られる（詳しい結果は第 4 章で示す）．それに対して，脇道を利用することで全てのエージェントが各々の目標状態へ辿り着くことができるよう

に改善することが求められる．

## 3.2 報酬の受容のモデルに内在する問題

### 3.2.1 問題

マルチエージェント学習ではタスク割り当ての観点を含めて学習させるように環境設計されることは少なくない．そういった環境のモデルにおいても，従来のエージェントは外部から与えられる報酬をそのままの報酬値として利用するが，目標状態が複数ある問題では容易に局所政策の獲得に陥る要因になり得る．

図 3.5 に示すのは前節でも述べた二体のエージェントが同じ旗の位置に到達することで両者に報酬が与えられる問題であるとする．設定として上のほうにある旗に到達して与えられる報酬の方が大きく，大局的な最適政策である問題とする．どちらの報酬を獲得する機会も同程度にあり，それらへ向かう政策を学習する期間も十分にあれば高い報酬の得られる目標へ向かうことが必然である一方で，報酬の与えられる目標状態の間に到達確率の偏りがある場合，局所政策の獲得に陥る．これは学習できてしまいさえすれば確実に到達できるような目標状態に対しても，それまでの到達確率が低く学習機会が少なければ適切に学習することが難しいという点で大きな問題である．この問題に対して，全ての報酬の獲得難易度（目標状態への到達確率）に見合った報酬値の設計するというアプローチは，それぞれの報酬獲得の難易度が他の報酬の大きさやエージェントの位置関係など様々な要因によって影響を受けるため難しい．また，対象問題が報酬値に相当するものを初めから持っている場合は，報酬設計の変更というアプローチは取りにくい．一方，学習中だけエージェントが独自に報酬を解釈し直すことで，可能な限り獲得報酬の最大化を目指すというアプローチは可能である．例えば，現実世界で商品の値段は誰に対しても一定であるが，それを見て高いと感じるか，安いと感じるかは各人の状況に依るということがあるように，エージェントが置かれた状況によって報酬値を独自に解釈し直す．そこで，与えられた報酬値をそのままの大きさとして受け取るのではなく，例えば獲得報酬の平均値と比べてどうかという観点でみると，高い報酬の存在によって上げられた獲得報酬の平均値は低い報酬への関心を下げ，相対的に高い報酬の探索に対するモチベーションの上昇に貢献するのではないかと考えられる．それによって図 3.6 のように大局的な最適政策を獲得することが期待できる．

### 3.2.2 例題

ここでは，複数の到達難易度（到達確率）の異なる目標状態（報酬）が存在し，それに対する学習によって結果として全体にとって最適でない（全エージェントが真に獲得報酬

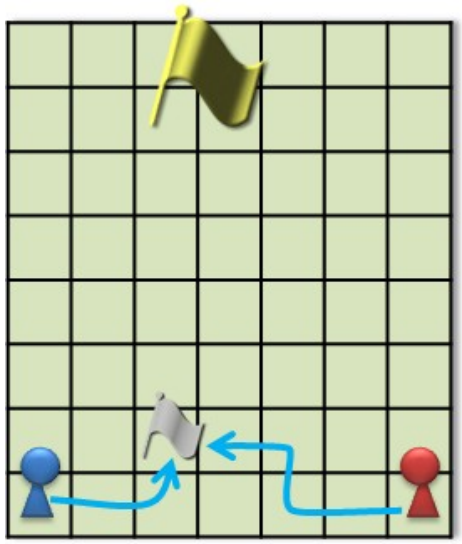


図 3.5　報酬の受容のモデルに内在する問題における局所政策獲得

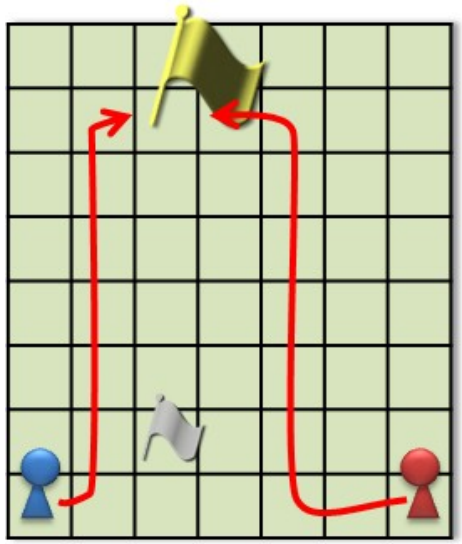


図 3.6　報酬の受容のモデルに内在する問題の解消

を最大化できない）政策を獲得する可能性のある例題を扱う．マルチステップ複数報酬問題では，エージェントは局所的な政策に陥りやすい．報酬獲得が容易だが報酬の低いタスクの組（陥りやすい局所政策）と，報酬獲得が困難だが報酬の高いタスクの組が存在するため，大局的な最適政策の獲得は困難となる．報酬の高いタスクの達成の難易度は，その座標が相手の達成の難易度の低いタスクの目標座標と同じであることにより，いっそう高められている．

マルチステップ複数報酬問題は図 3.7 のような二次元離散座標で表される．二体のエージェント A，B それぞれの初期座標は StartA，StartB，目標座標は互いに共通の二箇所

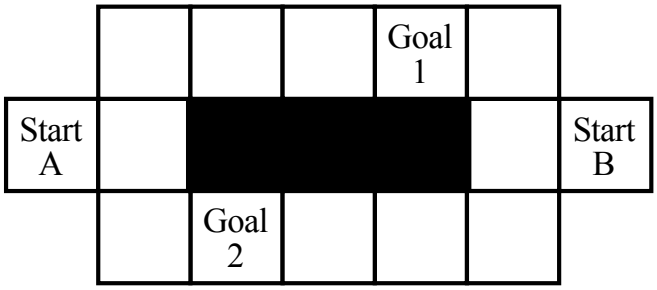



図 3.7　マルチステップタスク割り当て問題

あり Goal1，Goal2 として表されている．この環境におけるエージェントの学習目標はそれぞれの初期座標から同時に出発して両方のエージェントがいずれかの目標座標へ到達することである．

エージェントは 1 ステップに一つの行動を実行する機会が与えられる．このとき 1 ステップにおいてエージェント A，エージェント B の順番に行動し，全てのエージェントの行動の結果，エージェント毎に別々の報酬が環境の応答として与えられる．報酬は二種類あり，一つ目はエージェントが目標座標へ到達したステップに与えられるゴール報酬で，二つ目はそれ以外のステップに与えられる通常報酬である．ゴール報酬 $r_G$ は表 3.1 に示すようにエージェントの到達した目標座標によって定められており，目標座標に到達できなかった各ステップにおける報酬 $r_N = -0.01$ とする．表 3.1 は各エージェントの到達した目標状態の組とその時に各エージェントに与えられる報酬の組を示している．報酬の組の第一項はエージェント A の報酬，第二項はエージェント B の報酬を表す．また全てのエージェントがゴール報酬を得ると，エージェントは初期座標に戻され，引き続き学習を行い定められた回数（例えば `MAX_EPISODE`= 1000）だけ繰り返す．この繰り返し一回をエピソード（episode）と呼び，ゴール報酬は 1 エピソードの間では各エージェントに対して一回だけ，目標座標へ到達したステップにおいてのみ与えられる．また，初期座標に戻された状態から一定ステップ（例えば `MAX_STEP`= 100）経過しても全てのエージェントがゴール報酬を得ることができないとき，そのエピソードは強制的に打ち切られ，次のエピソードへ移行する．エピソードの開始時には必ずエージェントは初期座標に戻される．

表 3.1　到達した目標状態と与えられる報酬の対応

| | | agent B | |
|---|---|---|---|
| | | Goal 1 | Goal 2 |
| agent A | Goal 1 | - | $(10, 10)$ |
| | Goal 2 | $(5, 5)$ | - |

### 3.2.3 強化学習エージェントの挙動と改善の要件

強化学習エージェントが上記の設定の下で学習を行った場合，エージェント A が Goal2，エージェント B が Goal1 へ辿り着き，両者最低の報酬である 5 を得る政策を学習する（詳しい結果は第 5 章で示す）．それに対して，全てのエージェントが最高の報酬である 10 を得る政策を学習できるように改善することが求められる．与えられた報酬値をそのまま受け取るのではなく別に解釈することによる改善が考えられる．

## 3.3 単一政策の保持のモデルに内在する問題

### 3.3.1 問題

マルチエージェント環境では報酬が複数のエージェントの状況に応じて異なるものとして与えられる状況が多くなる．そうした前提では，大局的な最適政策を見つけ出す前に他の局所的な政策に陥ってしまう可能性が大きくなることが考えられる．エージェントが複数の報酬に対して個別に報酬最大化を目指すとき，エージェントは様々な報酬を受け取った経験から学習した期待できる報酬値の高い行動を取りながら学習するため，低い報酬のある状態付近を探索しなくなる．そのような理由から，自身の報酬が低くても他方のエージェントの報酬が高いといったようにエージェントが得る報酬間にトレードオフ関係がある場合，全体として最適な政策の獲得が阻害される．このように報酬の組み合わせが複数ある状況で大局的最適政策を確実に見つけるためにはただ一つの政策を保持するだけでは不十分である．また，トレードオフ関係にある場合，安直に一つの政策に絞ることが適切でないこともあり，最終的な候補を全て残す意味合いにおいてはそれら全てに対する政策を獲得できることも望ましい．

図 3.8 に示すのは前節でも述べた二体のエージェントが同じ旗の位置に到達することで両者に報酬が与えられる問題であるとする．設定として上のほうにある旗に到達して与えられる報酬の方が大きく，大局的な最適政策である問題とする．どちらの報酬を獲得する機会も同程度にあり，それらへ向かう政策を学習する期間も十分にあれば高い報酬の得られる目標へ向かうことが必然的であるが，一方で到達の確率に偏りがあって報酬を獲得する機会もそれに向かう政策を学習する期間も十分にない報酬が存在する場合，局所政策を獲得することに陥る．一方で，図 3.9 に示すように，エージェントに複数の政策を保持することを許せば，エージェンとはそれまで得た政策を破壊することなく保持したままの状態で他の政策を学習することができる．また，保存した政策を利用することで確率的に探索する場合よりも効率よく別の政策を探索できることが考えられる．

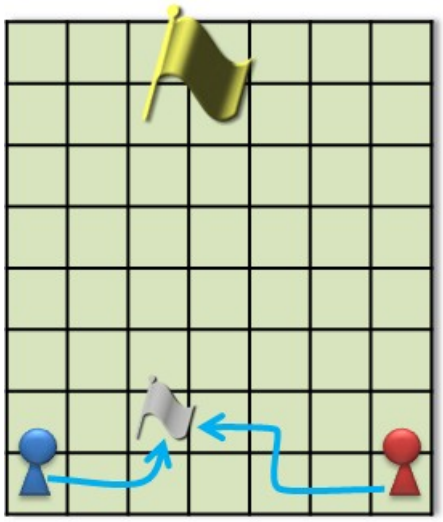


図 3.8　単一政策の保持のモデルに内在する問題における局所政策獲得

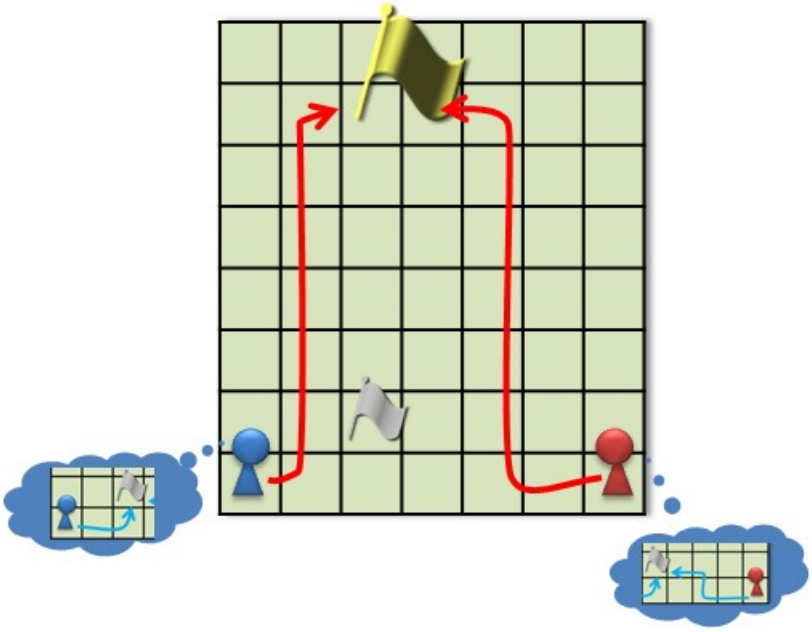


図 3.9　単一政策の保持のモデルに内在する問題の解消

### 3.3.2　例題

ここでは，複数の到達難易度（到達確率）の異なる目標状態（報酬）が存在し，それに対する学習によって結果として全体にとって最適でない（全エージェントが真に獲得報酬を最大化できない）政策を獲得する可能性があり，前述したマルチステップタスク割り当て問題と似た問題であるが，エージェントの報酬間にトレードオフ関係がある例題を扱う．

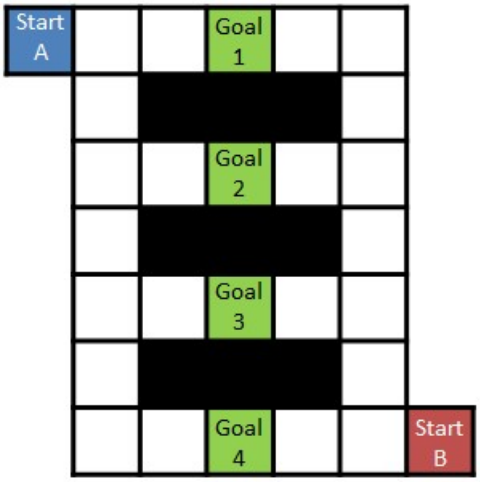



図 3.10　マルチステップ 4 タスク問題

マルチステップ 4 タスク問題は図 3.10 のような二次元離散座標で表される．二体のエージェント A，B それぞれの初期座標は StartA，StartB，目標座標は互いに共通の四箇所あり Goal1，Goal2，Goal3，Goal4 として表されている．この環境におけるエージェントの学習目標はそれぞれの初期座標から同時に出発して両方のエージェントがいずれかの目標座標へ到達することである．

エージェントは 1 ステップに一つの行動を実行する機会が与えられる．このとき 1 ステップにおいてエージェント A，エージェント B の順番に行動し，全てのエージェントの行動の結果，エージェント毎に別々の報酬が環境の応答として与えられる．報酬は二種類あり，一つ目はエージェントが目標座標へ到達したステップに与えられるゴール報酬で，二つ目はそれ以外のステップに与えられる通常報酬である．ゴール報酬 $r_G$ は表 3.2 に示すようにエージェントの到達した目標座標によって定められており，目標座標に到達できなかった各ステップにおける報酬は $r_N = -0.01$ とする．表 3.2 は各エージェントの到達した目標状態の組とその時に各エージェントに与えられる報酬の組を示している．報酬の組の第一項はエージェント A の報酬，第二項はエージェント B の報酬を表す．報酬の組み合わせが前の例題よりも増加しているがその構成は単純であり，エージェントは初期座標より遠い目標座標へ到達するほど自身の獲得できる報酬は高い．ただし報酬は他のエージェントの影響で上下し，具体的には互いの到達した座標までの距離（一段階遠いゴールへは距離が 1 大きいとする）を比べて相手より遠い分だけ報酬が増え，相手より近い分だけ報酬が減る．また全てのエージェントがゴール報酬を得ると，エージェントは初期座標に戻され，引き続き学習を行い定められた回数（例えば `MAX_EPISODE`= 1000）だけ繰り返す．この繰り返し一回をエピソード（episode）と呼び，ゴール報酬は 1 エピソードの間では各エージェントに対して一回だけ，目標座標へ到達したステップにおいてのみ与え

られる．また，初期座標に戻された状態から一定ステップ（例えば MAX_STEP= 100）経過しても全てのエージェントがゴール報酬を得ることができないとき，そのエピソードは強制的に打ち切られ，次のエピソードへ移行する．エピソードの開始時には必ずエージェントは初期座標に戻される．

表 3.2　到達した目標状態と与えられる報酬の対応

| | | AgentB | | | |
|---|---|---|---|---|---|
| | | Goal 1 | Goal 2 | Goal 3 | Goal 4 |
| AgentA | Goal 1 | - | (2,14) | (3,9) | (4,4) |
| | Goal 2 | (6,18) | - | (8,8) | (9,3) |
| | Goal 3 | (11,17) | (12,12) | - | (14,2) |
| | Goal 4 | (16,16) | (17,11) | (18,6) | - |

### 3.3.3　強化学習エージェントの挙動と改善の要件

強化学習エージェントが上記の設定の下で学習を行った場合，いくつものゴールが最も遠いゴールまでの道のりの付近に設定されているため，容易に最適解を得ることはできない（詳しい結果は第 6 章で示す）．それに対して，局所政策（それよりも優れた政策がある政策）を避けて全てのエージェントが自身の報酬を最大化しようとしたとき陥るはずの均衡である報酬の組 $(16, 16)$ が獲得できることが一つの目標である．またエージェント間の報酬にトレードオフ関係を持つ問題において局所政策ではないという意味で，エージェント間でのパレート最適となる（トレードオフを成している）報酬の組 $(6, 18), (11, 17), (17, 11), (18, 6)$ を獲得できることも一つの望ましい結果である．

# 第 4 章

# 学習進度に基づく学習速度の調整による競合回避

本章では，各エージェントの現在の行動がどれだけ確定的であるのか（行動が確定的であると，他のエージェントに譲る行為ができなくなる）を評価するために，初期状態から目標状態まで遷移した状態における行動選択確率のエントロピーを平均することで「学習進度」を定量化する．続いて，エージェント間で学習進度の差が大きくなり過ぎないようにするため，学習進度を通信することによって共有しながら，自身の学習速度を調整する手法を提案する．上記の競合回避メカニズムはエージェントの強化学習（ここではQ学習）の枠組みに組み込まれる．これはマルチエージェント強化学習のモデル全体から見ると，図 4.1 の黄色の部分に当たる．具体的には，エージェントはエピソードごと（目標状態に到達したとき，あるいは既定回数の行動を実行しても到達しなかったとき）に他のエージェントと学習進度を共有し，その情報を基に次のエピソードで学習速度を調整するかどうかを自律的に決定する．本研究では，学習速度の調整方法としていくつかの方法を考案し，それぞれの方法が競合回避に対して，どのようにかつどれだけ貢献できるかを網羅的な調査を行った．

本研究で注目しているマルチエージェント環境における競合は，エージェント間の学習進度に差が開くことで，学習が遅れているエージェントがさらに学習が進められないという状況である．その競合を回避するためには，エージェントは他のエージェントの学習進度に応じて学習の方針を変更する機構をもつ必要がある．そこで本章では既存研究である学習進度の定量化手法と，それを利用した競合回避手法を提案する．

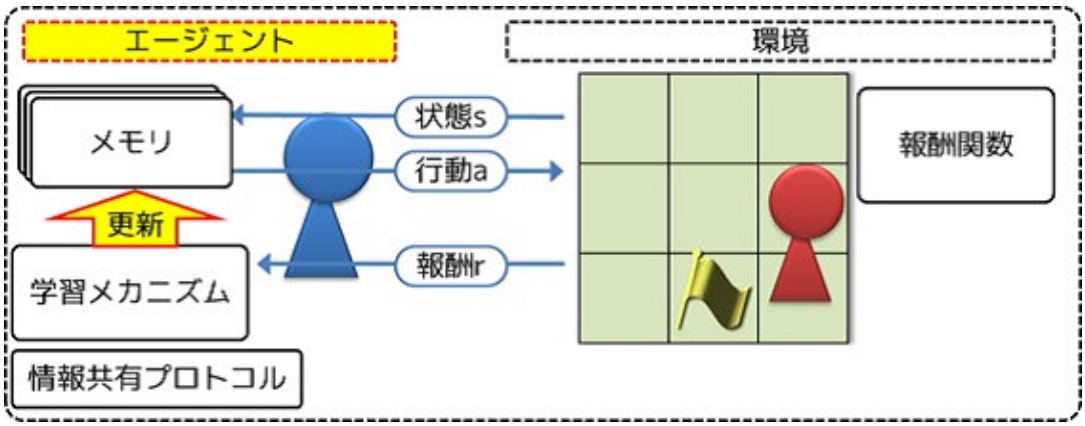



図 4.1　本章が対象とする領域

## 4.1　学習進度の定量化

### 4.1.1　情報エントロピーを用いた学習進度の定量化

情報理論に基づく定量化手法として，エージェントの持つエントロピー（entropy）を用いてエージェントの学習過程を評価する手法が提案されている [1] [24]．Q 学習エージェントにおいて，ある状態 $s$ におけるエントロピーは，状態 $s$ において取り得る全ての行動 $\mathcal{A}$ と各行動 $a$ が選択される確率 $\pi(s,a)$ を用いて次式により求められる．

$$H(\pi(s,\mathcal{A})) = -\sum_{a\in\mathcal{A}} \pi(s,a)\log\pi(s,a) \tag{4.1}$$

この手法に基づく学習進度の定量化であるエントロピー $H$ とは行動の曖昧さを表すスカラー量であり，ある状態で取りうる行動が唯一ならば，その行動が選択される確率は 1 であり，その状態におけるエージェントのエントロピーは式 (4.1) より 0 となる．一方で，取りうる行動が複数存在し，それぞれが等しい確率で選択されるとき，エントロピーは最大となる．なお，行動が複数あっても，行動が選択される確率に偏りがあるならばエントロピーが低くなる．学習進度という観点から言い換えると，エージェントの意思決定が不確定（例えば学習初期）であるほどエントロピーは高い値を示し，学習が進み行動が確定的になるとエントロピーは低い値を示す．

### 4.1.2　エピソードに基づくエントロピー

本研究では，エージェントが 1 エピソード終えた時点で持つ状態-行動価値を基に，エピソード毎に求められるエントロピーを提案する（全ての状態集合におけるエントロピーでないことに注意されたい）．なぜならば，学習が進むにつれてエージェントが観測する

状態の分布が時間とともに変化し，重要なのは現時点で頻繁に観測する状態についてのエントロピーと考えられるからである．具体的に，エージェントはそのエピソードで経験した状態 $s$ をかかったステップ $t$ の数だけ記憶（$\mathcal{S}_{episode} = \{s_1, s_2, s_3, \ldots, s_t\}$）し，各状態におけるエントロピーを式 (4.1) により求めた後，そのエピソードで行動を選択した回数，つまり記憶した状態数で割り，エピソードに基づくエントロピーとする．これは 1 エピソードの間のに観測した全状態の平均エントロピーであり，エージェント X のエントロピー $H(X)$ は次の式により導かれる．

$$H(X) = \frac{\sum_{s \in \mathcal{S}_{episode}} H(\pi(s, \mathcal{A}))}{|\mathcal{S}_{episode}|} \tag{4.2}$$

### 4.1.3 予備実験：例題におけるエントロピーの観察

狭路すれ違い問題を前述の設計に基づいて計算機上に実装し，実験により得た成功時と競合時の試行における二体のエージェントのエントロピーの変化を観察する．エージェントは Q 学習，ボルツマン分布に基づく選択により学習し，環境，エージェントの設定は前章で示したものと同じものを適用し，表 4.1 にまとめる．$*$ 印のパラメータはどの実験においても共通のため，以後は省略する．

表 4.1 予備実験環境とパラメータ

| | | |
|---|---|---|
| 環境 | a（図 3.4） | |
| エージェント数 | 2 | |
| MAX_STEP | 100 | |
| MAX_EPISODE | 500 | |
| 学習率初期値 $\alpha_0$ | 0.2 | $*$ |
| 割引率初期値 $\gamma_0$ | 0.95 | $*$ |
| ゴール報酬 $r_G$ | 5 | $*$ |
| 通常報酬 $r_N$ | $-0.01$ | $*$ |

図 4.2，4.3 にその結果を示す．グラフはそれぞれ縦軸がエントロピー，横軸がエピソード数，二つの点線がそれぞれエージェント A のエントロピー $H(A)$，エージェント B のエントロピー $H(B)$，＋（プラス記号）のプロットは 1 エピソードが終了したステップを表しており値は右軸に記されている．なお，エントロピーは過去 10 エピソードの移動平均を取ったものをプロットする．成功時のエントロピーというのは最終的に両方のエージェントが目標座標へ到達できるように学習した（＋ が減少しており，両エージェントがゴール報酬を得ることができている）際のグラフであり，これをみると，エントロピーの

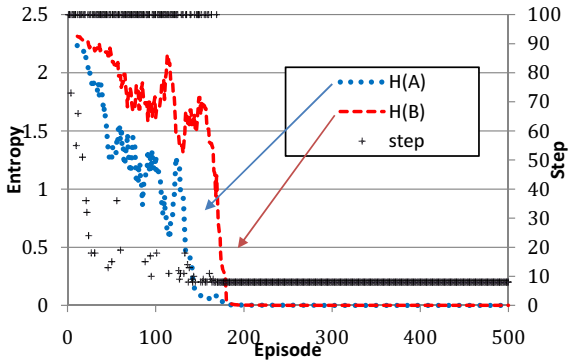



図 4.2　成功時のエントロピー

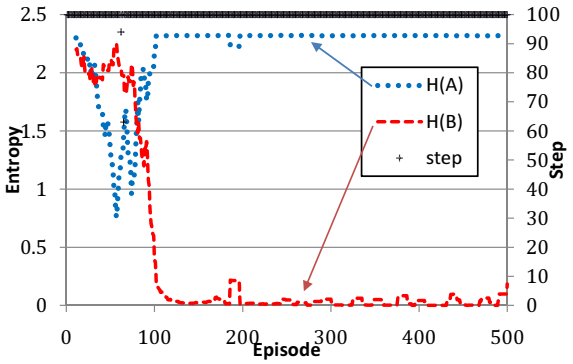



図 4.3　競合時のエントロピー

定義通りに両エージェントのエントロピーが減少している．一方，競合時のエントロピーは右から出発するエージェント B だけが目標に到達することができた（+ がずっと **100** のままでいずれかのエージェントがゴールに到達していない）際のグラフであり，これをみると，片方のエージェント B の学習のみが進みエントロピー $H(B)$ が減少しているなかで，他方のエージェント A のエントロピー $H(A)$ はいつまでも下がっていない．この結果より，エージェント B のエントロピーが下がり行動が確定的になると，一方のエージェント A の学習が進まなくなる様子がみられることから，提案したエピソードに基づくエントロピーは学習進度の変化を表すことができていることがわかる．

## 4.2　提案１：学習進度に着目した競合回避エージェント

### 4.2.1　概要

前章で述べたように，競合が生じる原因は，あるエージェントの学習が進み行動が決定的になることで，他方のエージェントの学習を阻害するためである．そこで，まず二体エージェント問題を仮定し，二体のエージェントの学習進度の差を表すエントロピーの差に着目し，その差が大きくなりすぎないようにバイアスをかけることで競合を回避する手法を提案する．具体的には，学習進度にある程度の差が生まれたとき，エージェントの学習速度を一時的に変更する．学習が進んでいるエージェントの学習速度を下げれば，相対的に他方の学習が早まり，また学習が遅れているエージェントの学習速度を上げることでも同様に学習進度の差が縮まることが期待できる．学習進度の定量化として前述のエピソードに基づくエントロピーを用い，その定義より本研究ではエントロピーが減少することを学習の進行であるとし，エントロピーが低いほど学習が進んでいると捉える．

### 4.2.2 アーキテクチャ

提案手法におけるエージェントは，図 4.4 に示すように，他のエージェントと情報を共有するための通信機能，一定の条件下で学習速度を切り換えるメカニズム，エントロピーの計算機能とエントロピーを記憶する機構を有する．また，エージェントの学習速度の変更は中央集権的に行うのではなく，エージェントが受信した情報から独自に判断を下す．

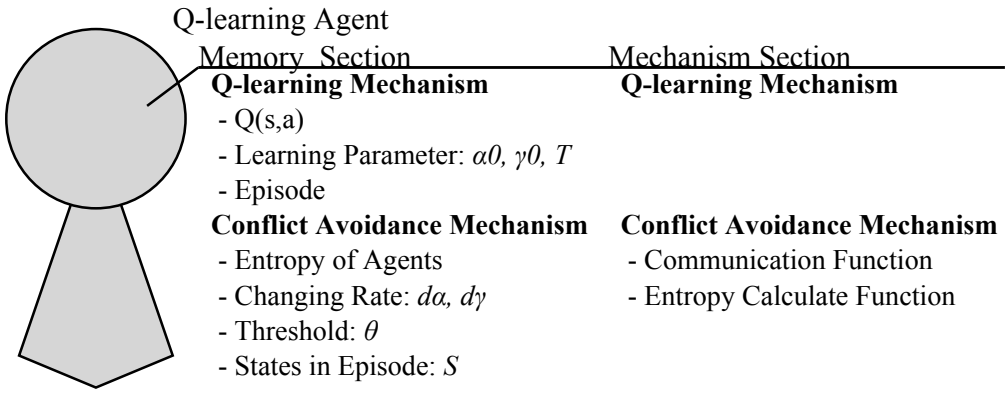



図 4.4 エージェントアーキテクチャ

### 4.2.3 メカニズム

ここで学習速度を切り換えるメカニズムは次の四段階からなる． 図 4.5 に示すように，(Step 1) エピソード終了直後にエピソードに基づくエントロピーを算出し，(Step 2) 他エージェントとエントロピーを共有し，(Step 3) 両エージェントのエントロピーの差がある閾値を超えた時に，(Step 4) 一方のエージェントの学習パラメータを変更することで学習速度を変更する．具体的に，(Step 3) は次の条件式で表される．

$$|H(A) - H(B)| > \theta \tag{4.3}$$

ここで閾値となる $\theta$ は適当な値に設定する．一方，(Step 4) の学習速度変更は，$H(X) > H(Y)$ を満たしているとき，(del) 学習が進んでいるエージェント Y の学習速度を下げる，あるいは (acc) エージェント X の学習速度を上げることが挙げられる．また，学習を遅くする方法として (a) 学習率 $\alpha$，(b) 割引率 $\gamma$ を下げる方法，同様に学習を速くする方法として (a) 学習率 $\alpha$，(b) 割引率 $\gamma$ を上げる方法を上げる．学習パラメータを上げることは，Q 学習における $\mathrm{Q(s,a)}$ の更新式である式 (2.1) にを変形した次式によれば，$\alpha$ を下げることは第二項である今回の結果の比重を小さくすることになり，学習を遅くできると

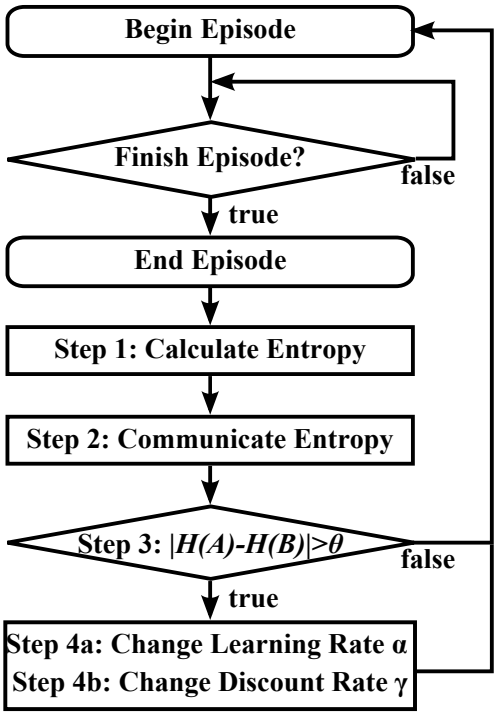



図 4.5　手法適用の流れ

言え，$\gamma$ を下げることによっても第二項が小さくなり同じことが期待できる．

$$Q(s,a) \leftarrow (1-\alpha)Q(s,a) + \alpha\left[r + \gamma \max_{a' \in \mathcal{A}'} Q(s',a')\right] \tag{4.4}$$

学習パラメータは Q 学習のパラメータの初期値 $\alpha_0$，$\gamma_0$ と，それらを変化させるための量 $d\alpha$，$d\gamma$ を用いて次式の通りに変更する．

$$\alpha \leftarrow \alpha_0 \times d\alpha \tag{4.5}$$

$$\gamma \leftarrow \gamma_0 \times d\gamma \tag{4.6}$$

以後，表記を簡単にするため，他エージェントとのエントロピーの差が $\theta$ より大きく式 (4.3) を満たしているとき，エントロピーが低く学習が進んでいるエージェントとエントロピーが高く学習が遅れているエージェントをそれぞれ「LowH エージェント」，「HighH エージェント」と表記する．当然，LowH または HighH エージェントはエントロピーの差が設定した閾値 $\theta$ より上回った場合だけ存在し，それ以外の場合において本手法はいかなるエージェントに対してもバイアスをかけることはない．最後に，本手法を **Algorithm 1(1)** にまとめる．大文字斜体で書かれた単語は関数，その他の大文字は定数を表しており，$Env$ は環境の略称でありドットに続く情報はエージェントではなく環境が保持している．ID はエージェントの識別符号を表している．

**Algorithm 1(1)** Conflict Avoidance Method based on Entropy

$\alpha \leftarrow \alpha_0 \quad \gamma \leftarrow \gamma_0$
$episode \leftarrow 0$
**while** $episode <$ `MAX_EPISODE` **do**
  $s \leftarrow Env.StartState(\mathrm{ID})$
  $step \leftarrow 0$
  **while** $step <$ `MAX_STEP` $ors \not\equiv Env.$`GOAL_STATE`$(\mathrm{ID})$ **do**
    $StoreState(s)$
    $a \leftarrow SelectAction(Q(\mathcal{S}, \mathcal{A}))$
    $DoAction(a)$
    $s' \leftarrow Env.State(\mathrm{ID})$
    $r \leftarrow Env.Reward(s')$
    $Q(s,a) \leftarrow UpdateQ(r, s', A', \alpha, \gamma)$
    $step \leftarrow step + 1$
    $s \leftarrow s'$
  **end while**
  $h \leftarrow CalculateEntropy(Q(s,a))$ (Step 1)
  $StoreEntropy(h)$
  $H($`OTHERS_ID`$) \leftarrow Communicate($`OTHERS_ID`$, h)$ (Step 2)
  **if** `CONDITON`(**Methods**) **then** (Step 3)
    $\alpha \leftarrow \alpha_0 \times d\alpha \quad \gamma \leftarrow \gamma_0 \times \mathrm{d}\gamma$ (Step 4)
  **else**
    $\alpha \leftarrow \alpha_0 \quad \gamma \leftarrow \gamma_0$
  **end if**
**end while**
**#define** `CONDITON`(`Method_del`)$LowEntropy($`OTHERS_ID`$, h)$
**#define** `CONDITON`(`Method_acc`)$HighEntropy($`OTHERS_ID`$, h)$

なお，Q 学習を用いたエージェントを扱う本稿では上記の方法を挙げるが，エージェントが他の強化学習手法により学習する場合，さらに別の方法で学習速度を変更する必要が出る可能性があることを補足しておく．

## 4.3 提案２：三体エージェント環境への拡張

### 4.3.1 概要

二体エージェント問題においては，いつでも特定のエージェント間の学習進度の差を，片方のエージェントの学習パラメータを変更することで相対的に調整したが，エージェント三体以上の問題では，(1) どのエージェントとのエントロピーの差を基に，(2) どのエージェントが学習速度を変更するかが問題となる．そのため，本節では前節で提案した手法を三体以上に適用可能な一般的な形に拡張する．具体的には，前節で示したメカニズムの一部を変更し，全てのエージェントとエントロピーを共有することで，各エージェント自身が相対的にどのくらい学習が進んでいるのかを判断し，学習速度を変更する．

### 4.3.2 アーキテクチャ

提案２におけるエージェントは提案１の純粋な拡張であるため，基本的なアーキテクチャは提案１のエージェント（図 4.4）と同じである．それに加えて提案２におけるエージェントは全エージェントに情報を送り，受信し，そのエントロピー全てを記憶，順序づけする機能が必要である．送られてきた情報がどのエージェントからのものか識別する必要はないため，送受信の量はエージェント数が $N_a$ のとき高々 $(N_a - 1)^2$ である．

### 4.3.3 メカニズム

具体的には前節で示した (Step 3) を変更し，問題 (1) については単純に全てのエージェントと一対一で通信し，それぞれのエージェントとのエントロピーを比較するものとし，問題 (2) については少なくとも一体のエージェントとの間で式 (4.3) を満たし，かつ学習が遅れている（進んでいる）エージェントの中の (I) 全てのエージェント，あるいは (II) 最も学習が遅れている（進んでいる）エージェントの学習速度を上げる（下げる）．手法 (I) は最も学習が進んでいる（または遅れている）エージェントに対して，他の条件を満たすエージェントの全てが学習進度の差を減らすようなバイアスてあり，一方，手法 (II) は最も学習が遅れているエージェントのみが，その他の全てエージェントに対して学習進度の差を減らすようなバイアスである．どちらの手法においても仮に上手く学習進度の差が縮まり，次第に逆転すれば，それぞれのバイアスの対象が変更されるため結果として同じような効果が得られると期待できるが，適用されるタイミングがエージェント毎に異なるため，それぞれバイアスが学習進度の差どう縮めるかという点で異なる効果が期待できる．以後，提案１と同様に表記を簡単にするため，少なくとも一体のエージェントとのエントロピーの差が $\theta$ より大きく式 (4.3) を満たしているとき，エントロピーが低く学習が

進んでいるエージェントとエントロピーが高く学習が遅れているエージェントをそれぞれ「LowerH エージェント」,「HigherH エージェント」と表記し，その中で最もエントロピーが低いまたは高いエージェントをそれぞれ「LowestH エージェント」,「HighestH エージェント」と称する．つまり，LowerH と HigherH エージェントは手法 (I)，LowestH と HighestH エージェントは手法 (II) において学習速度の変更の対象となるエージェントを指す．

図 4.6 に三体エージェントにおける手法の適用例を示す．グラフは縦軸にエントロピーの大きさ（学習進度），横軸に二つの状況における各エージェントのエントロピーを並べている．その下の表に，それぞれの手法（(I) と (II) の手法においてそれぞれ学習速度を (del) 下げるか，(acc) 上げるかの四つの組み合わせ）において，学習速度が変更されるエージェントの下方に下矢印（遅くする）または上矢印（速くする）を記した．全エージェント中でエントロピーが中間に位置するエージェント B は，状況 1 のとき手法 (I-acc) ではエージェント A がいるため学習速度を上げないが，手法 (II-acc) ではエージェント A の存在にかかわらず条件式 (4.3) を満たすから学習速度を上げる．状況 2 についても同様である．

最後に，本手法を **Algorithm 1(2)** にまとめる．大文字斜体で書かれた単語は関数，その他の大文字は定数を表しており，*Env* は環境の略称でありドットに続く情報はエージェントではなく環境が保持している．ID はエージェントの識別符号を表している．

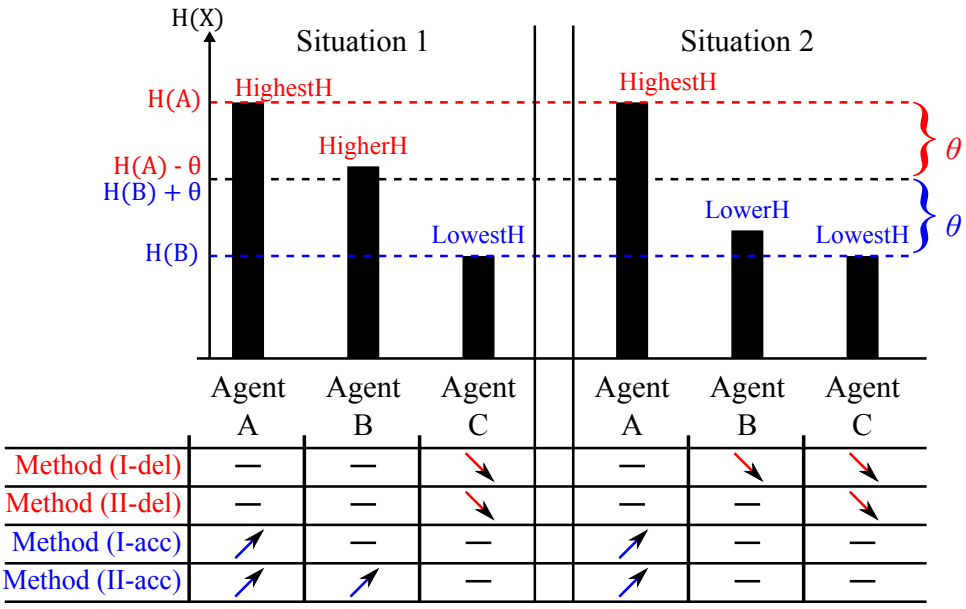



図 4.6　手法適用例

**Algorithm 1(2)** Conflict Avoidance Method based on Entropy

$\alpha \leftarrow \alpha_0 \quad \gamma \leftarrow \gamma_0$
$episode \leftarrow 0$
**while** $episode <$ `MAX_EPISODE` **do**
  $s \leftarrow Env.StartState(\mathrm{ID})$
  $step \leftarrow 0$
  **while** $step <$ `MAX_STEP` $ors \not\equiv Env.GoalState(\mathrm{ID})$ **do**
    $StoreState(s)$
    $a \leftarrow SelectAction(Q(\mathcal{S}, \mathcal{A}))$
    $DoAction(a)$
    $s' \leftarrow Env.State(\mathrm{ID})$
    $r \leftarrow Env.Reward(s')$
    $Q(s,a) \leftarrow UpdateQ(r, s', A', \alpha, \gamma)$
    $step \leftarrow step + 1$
    $s \leftarrow s'$
  **end while**
  $h \leftarrow CalculateEntropy(Q(s,a))$ (Step 1)
  $StoreEntropy(h)$
  **for each agents**
    $H($`OTHERS_ID`$) \leftarrow Communicate($`OTHERS_ID`$, h)$ (Step 2)
  **if** `CONDITION`(**Mthods**) = **True then** (Step 3)
    $\alpha \leftarrow \alpha_0 \times d\alpha \quad \gamma \leftarrow \gamma_0 \times \mathrm{d}\gamma$ (Step 4)
  **else**
    $\alpha \leftarrow \alpha_0 \quad \gamma \leftarrow \gamma_0$
  **end if**
**end while**
**#define** `CONDITION(Method_I-del)` $LowestEntropy($`OTHERS_ID`$, h)$
**#define** `CONDITION(Method_II-del)` $LowerEntropy($`OTHERS_ID`$, h)$
**#define** `CONDITION(Method_I-acc)` $HigherEntropy($`OTHERS_ID`$, h)$
**#define** `CONDITION(Method_II-acc)` $HighestEntropy($`OTHERS_ID`$, h)$

## 4.4 実験 1 : 二体エージェント

### 4.4.1 実験内容

提案手法の有効性を検証するため，前述の狭路すれ違い問題（図 4.7）を計算機上に実装し，以下の表 4.2 に示すケースに別けて実験することで，それぞれの方法の効果の違いを明らかにする．縦軸はどんな方針の手法であるか表しており，横軸はどちらの学習パラメータを変更するかを表している．さらに，状態空間が大きく，より競合が起こりやすい難易度の高い問題環境 b（図 4.8）に対しても手法が有効であるか実験する．

表 4.2　実験 1 における実験ケース

| | 学習率 $\alpha$ を変更 | 割引率 $\gamma$ を変更 |
|---|---|---|
| LowH エージェントの学習パラメータを下げる | ケース 1a | ケース 1b |
| HighH エージェントの学習パラメータを上げる | ケース 2a | ケース 2b |

### 4.4.2 評価指標とパラメータ設定

評価は一定エピソード（MAX_EPISODE）の学習後，行動選択手法をグリーディ選択手法（greedy selection）に変更し，つまり $Q(s,a)$ が一番大きい最良の行動のみを選択させたときに，全てのエージェントが目標状態へ到達できるかどうかで，学習が成功であったかどうか判断する．実際には，同じ条件のもとで乱数シードを変え，100 回の実験の平均成功率を計算する．なお，$\theta$ はエントロピーの取りうる値，0 から理論的な最大値（5 行動の選択確率が等確率のとき $5 \times (-\Sigma(1/5)\lg(1/5)) \simeq 2.32$）より，その範囲の値に定める．また $d\alpha$，$d\gamma$ は学習率 $\alpha$，割引率 $\gamma$ が取る範囲より，それぞれ $[0 \leqq \alpha_0 \times d\alpha, \gamma \times d\gamma \leqq 1]$ に違反しない正の実数値とする．そして，いくつかの $\theta$ と $d\alpha, d\gamma$ を組み合わせて実験し，それぞれの成功率を比較する．また，二つの実験環境 a，b のパラメータは以下のように設定する．実験環境 b は状態数の増加に伴い実験期間を少し長く設定する．

### 4.4.3 実験結果

#### ケース 1a，1b

図 4.9，4.10 にケース 1a について，$\theta$，$d\alpha$ を変更した実験の結果を示す．縦軸は成功率，横軸は $d\alpha$，各ラインは同じ $\theta$ の結果を繋いだものを示している．$d\alpha = 1.0$ のときは学習率を変更しないことと同義であり，通常の Q 学習での成功率にあたる．その通常の

表 4.3　実験 1 の各問題環境とパラメータ

| 環境 | a（図 4.7） | b（図 4.8） |
|---|---|---|
| エージェント数 | 2 | 2 |
| MAX STEP | 100 | 200 |
| MAX EPISODE | 500 | 1000 |

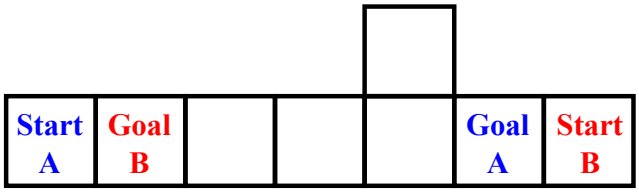



図 4.7　問題環境 a

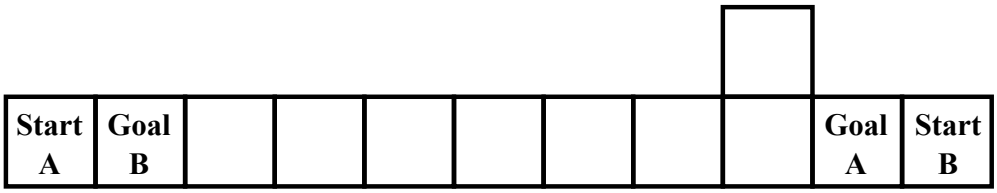



図 4.8　問題環境 b

Q 学習での成功率は問題環境 a では 90%，環境 b では 72% であり，言い換えると 10%，28% の確率で競合に陥るといえる．両方のグラフでは縦軸のスケールが異なることに注意されたい．結果をみると LowH（学習が進んでいる）エージェントの学習率 $\alpha$ を下げることにより成功率を高くすることができている．全体的にみると，$d\alpha$ が低い，つまり $\alpha$ を大きく下げ学習を極端に遅らせることで，より多くの競合が回避されている．この理由として，LowH エージェントの学習を遅らせることで，行動の偏りが本来よりも遅くなり，他のエージェントに対する学習の阻害が起こりにくくなるためであると考えられる．一方で $\theta = 2.1$ の場合，成功率が低い．この結果は，$\theta$ が大きい，つまり LowH エージェントの学習を遅らせる時期が学習進度に大きな差が生まれてからになると，$\alpha$ を下げることによる効果が得られにくいことを示している．ただ，環境 b では $\theta = 1.4$ 程度まで大きくても他の $\theta$ よりも大きな効果が見られる．以上の結果から，LowH エージェントの学習を学習率 $\alpha$ を下げて遅くすることで多くの競合が回避できる一方で，$\theta$ にかかわらず LowH エージェントの学習率の変更だけでは本質的に回避できないタイプの競合があることが明らかとなった．

次に，図 4.11，4.12 にケース 2 について，$\theta$，$d\gamma$ を変更した実験の結果を示す．縦軸は

成功率，横軸は $d\gamma$，各ラインは同じ $\theta$ の結果を繋いだものを示している．実験ケース 1a と同様に $d\gamma = 1.0$ のときは割引率を変更しないことと同義で，通常の Q 学習での成功率にあたり，その成功率は実験ケース 1a と同じである．結果をみると，環境 a，b ともに全体的な傾向は同じである．まず，成功率が 100% で全ての競合を回避できている設定が観測された．この設定は $\theta$ が高すぎたり，もしくは $\theta$ が低くかつ $d\gamma$ が小さいときを除く場合であることがわかる．ケース 1a と比較すると，$\theta = 2.1$ では同様に他の $\theta$ と比べて効果が低いことが見られるが，その結果はケース 1a の一番良い結果と比べてもそれに劣らず競合の回避に貢献している．また，単純にＬｏｗＨエージェントの $\gamma$ を大きく下げるほど結果が良くなるわけではなく，特に $\theta = 0.0$ のとき，$d\gamma$ が小さいと成功率を大きく低下させてしまう．以上の結果は $\gamma$ の変更が学習を遅くする以外に何らかの作用をもたらすことを示している．

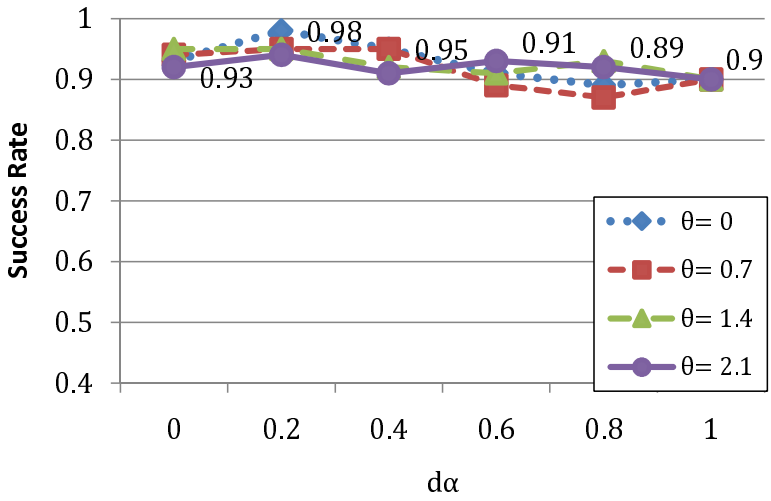



図 4.9　問題環境 a ケース 1a の成功率

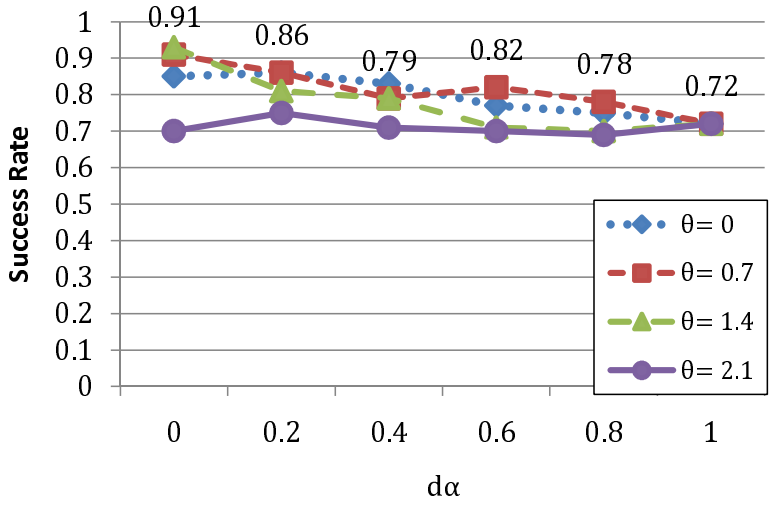



図 4.10　問題環境 b ケース 1a の成功率

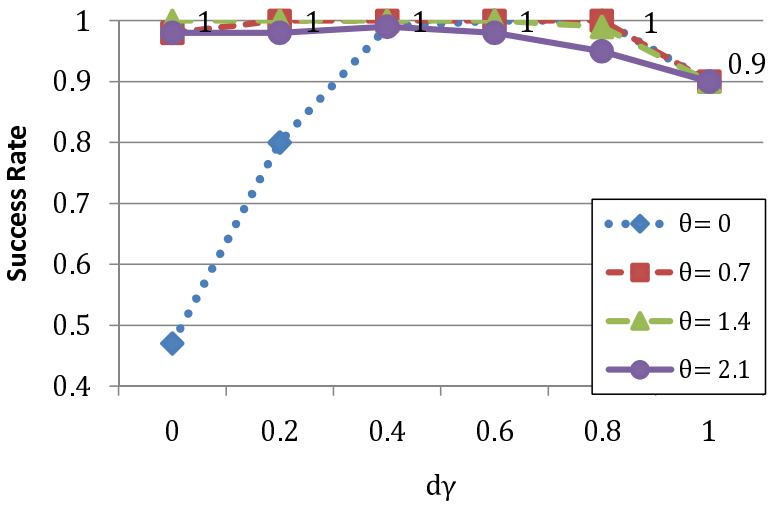



図 4.11　問題環境 a ケース 1b の成功率

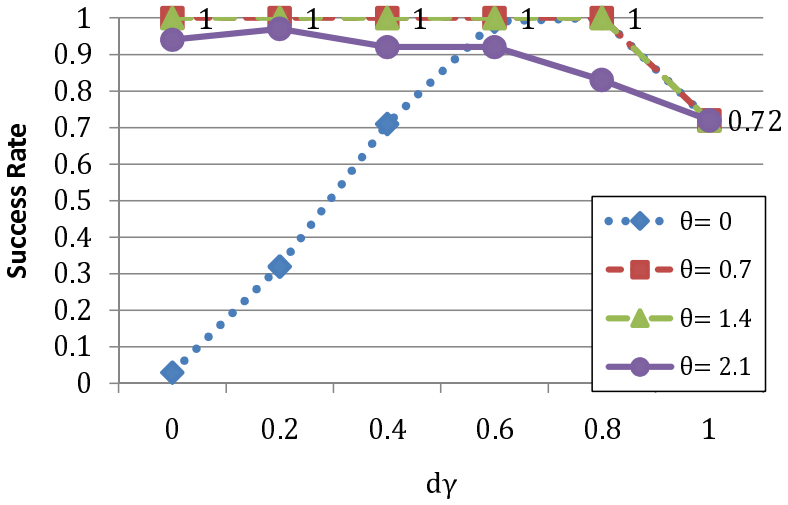



図 4.12　問題環境 b ケース 1b の成功率

**ケース** 2a，2b

図 4.13，4.14 にケース 2a について，$\theta$，$d\alpha$ を変更した実験の結果を示す．グラフは $d\alpha$ が取る範囲を除いてケース 1a と全く同じようにみることができる．結果をみるとＨｉｇｈ H（学習が遅れている）エージェントの学習率 $\alpha$ を上げることにより成功率を上げることができている．全体的にみると，$d\alpha$ が高い，つまり $\alpha$ を大きく上げるほど，より多くの競合が回避されている．この理由として，HighH エージェントの学習を早めることで，LowH エージェントの行動が確定的になり学習の阻害が起こる前に，ある程度まで学習が進められる可能性が増えるためと考えられる．ここでの「ある程度まで」とは LowH エージェントとのエントロピーの差が $\theta$ に近づくまでである．このようにエージェントはほとんどの場合，付かず離れずを繰り返し確率的に競合を回避するが，$\theta$ が大きいほど差を縮めるバイアスがかかる期間が少なくなるため競合回避が難しくなる．このことは環境 b の $\theta = 2.1$ において見られるが，$\theta = 0, 0.7, 1.4$ における結果に大きな差異が見られない．これはケース 1a の環境 b においても同様のことが言え，このことから，環境 b においては，根本的に適切な状態-行動価値の推定のための行動をとることができなくなるようなエントロピーの差が，1.4 から 2.1 の間に存在していることを示唆している．

次に，図 4.15，4.16 にケース 2b について，$\theta$，$d\gamma$ を変更した実験の結果を示す．グラフは横軸に示す $d\gamma$ の範囲が異なることを除いてケース 1b と同じように見ることができる．結果より，環境 a，b ともに $\theta = 0.0$，$d\gamma = 1.05$ とし HighH エージェントの割引率 $\gamma$ を上げることで成功率が 100% となることが観測された．環境 b では $d\gamma = 1.03$ をより大きくすることで目に見えて高い成功率が得られていることに加えて $\theta$ が低いほど競合回避に有効であることが明らかである．つまり，ケース 2b では単に $\theta$ を小さくかつ $d\gamma$ を大きく取ればよいため，ケース 1b と比較して，より有効な手段であるといえる．

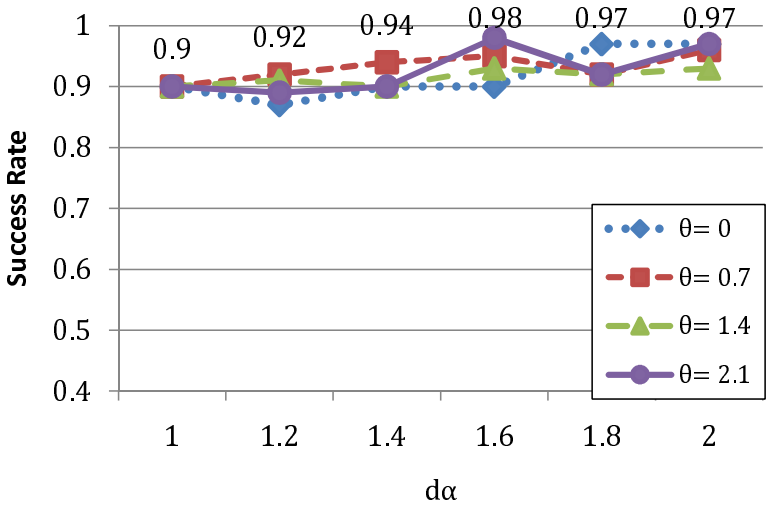



図 4.13　問題環境 a ケース 2a の成功率

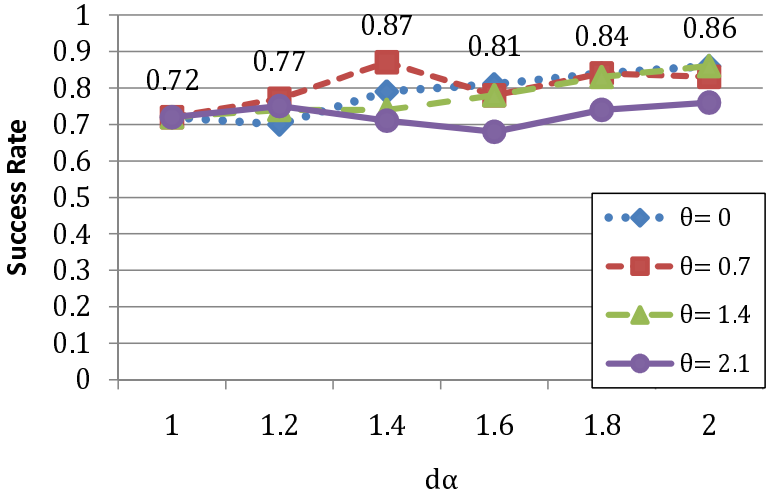



図 4.14　問題環境 b ケース 2a の成功率

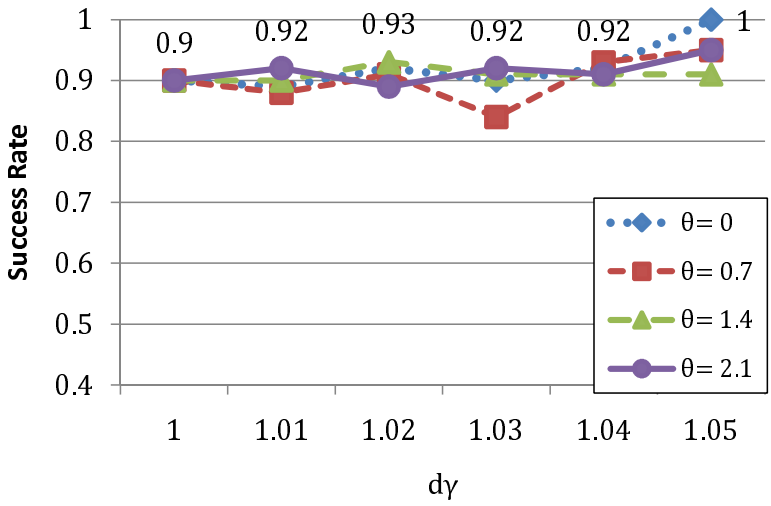



図 4.15　問題環境 a ケース 2b の成功率

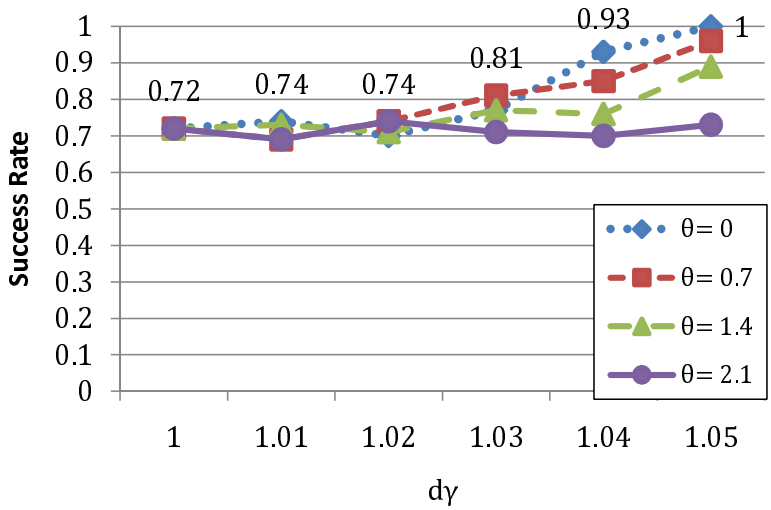



図 4.16　問題環境 b ケース 2b の成功率

### 4.4.4 考察

以上の実験より，LowH エージェントの学習を遅らせる場合でも，HighH エージェントの学習を早める場合でも，割引率 $\gamma$ を変更することが，多くの競合を回避することが明らかとなった．ここでは学習パラメータ変更が競合回避にもたらす効果をエピソードに基づくエントロピーと各エピソードの終了ステップ数の推移を提示し，考察する．

**学習の進行の傾向と競合回避の可能性**

まず，問題環境 b の各ケースにおいて全ての乱数シードの結果で平均したエピソード毎のエントロピーとエピソード終了ステップ数の推移を図 4.17〜4.20 に示す．グラフは横軸にエピソード数，左の縦軸に対して二体のエージェントのエントロピーをそれぞれ点線で表し，右の縦軸に対してエピソード終了ステップ数（全てのエージェントがゴール報酬を得られない場合は MAX_STEP）を + 記号で表している．これらは各ケースの代表一パラメータの結果であり，ケース 1a では $d\alpha = 0.0$，ケース 1b では $d\gamma = 0.6$，ケース 2a では $d\alpha = 2.0$，ケース 2b では $d\gamma = 1.05$，どのケースにおいても $\theta = 0.7$ である．ここに示される，ステップ数（+ 記号）の収束部分をみることで学習の早さを評価することができる．収束部分とは，グラフの変動がなくなるエピソードを指し，同等の結果（成功率 100%）を得た $\gamma$ を変更する二つのケース 1b と 2b では，約 200 エピソード LowH エージェントの $\gamma$ を下げるケース 1b の方が HighH エージェントの $\gamma$ を上げるケース 2b より早く学習可能であることが分かった．この理由は $\gamma$ の変更が学習に与える影響として後に考察する．ケース 1b，2a，2b は 1000 エピソードの時点でほぼ横ばいになっており，これ以上成功率が上がることはほとんどないといえる一方で，ケース 1a はこれ以降でまだ学習が進められたと言える．

もうひとつのグラフであるエントロピーを観察することで，各ケースがどのように学習を進めるか，傾向をつかむことができる．ケース 1b ではエージェント A のエントロピー $H(A)$（青線）が途中でエージェント B のエントロピー $H(B)$（赤線）を下回った後に，学習が完了する．このことからケース 1b ではエージェント B，A という順に学習が完了する（行動が確定的になる）傾向があることがわかる．ケース 1a と 2a を比較すると，早い段階でエージェント A のエントロピーを低くするケース 2a は，本来競合の原因である一方の方策の偏りを助長している結果といえ，成功率があまり上がらない要因である．別の調べによると，各乱数シード毎の手法適用前と後の学習の成功，競合を比べるとケース 1a と 2a では元々成功するはずの試行が競合になる場合があることが確認されており，$\alpha$ を変更することは本質的に競合を回避する能力を持っていないと考えられる．これは，学習率の変更が学習速度を変化させ，それによって競合が生じる確率が変わり好転する場合が多いものの，競合が起きるときは起きてしまうということを意味している．

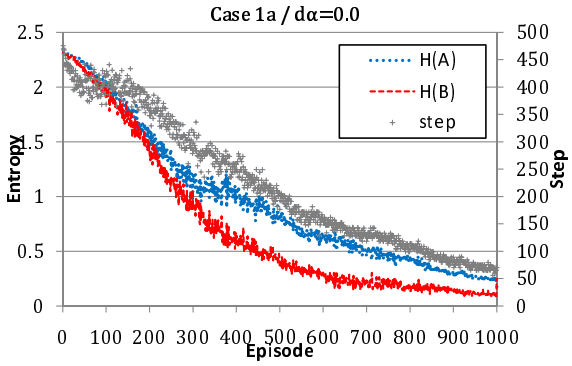



図 4.17　問題環境 b ケース 1a のエントロピー推移の平均

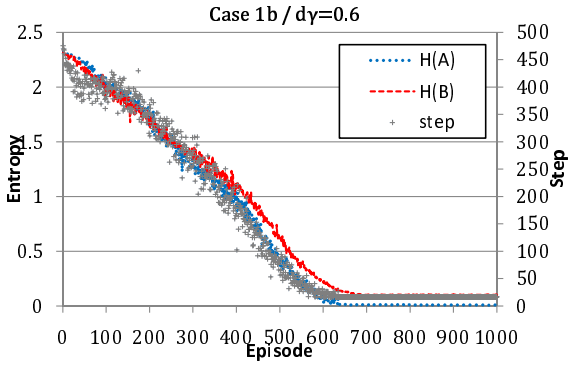



図 4.18　問題環境 b ケース 1b のエントロピー推移の平均

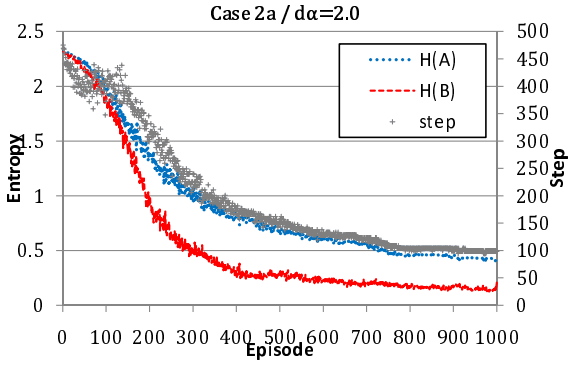



図 4.19　問題環境 b ケース 2a のエントロピー推移の平均

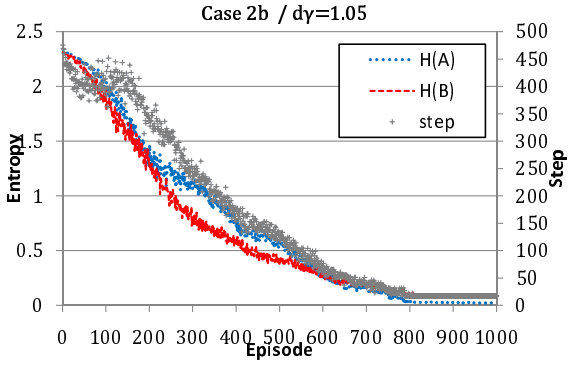



図 4.20　問題環境 b ケース 2b のエントロピー推移の平均

**学習率 $\alpha$ 変更の効果**

次に，各ケースにおける代表一シードのエントロピーの推移を観察する（図 4.21〜4.24）．同時に，エントロピーの差 $H(A) - H(B)$（紫の実線）と学習パラメータが変更される閾値 $\theta$（緑の実線）をみることで，実際に学習速度が変更されたエピソードを知ることができる．この例題においてケース 1a とケース 1b の具体的な狙いは，エージェント B の学習を遅くして，行動を確定的にしないことである．LowH エージェントの $\alpha$ を下げるケース 1a のグラフをみると，学習を遅くする機会がどちらのエージェントに対しても頻繁に現れているため，エージェント A，B 共に中々学習が進められない一方で，両エージェントは交互に学習を進めるが，ある時点でエージェント B がほぼ確定的な行動をとるようになり，B が価値推定したい状態を経験する可能性が減少すると，その先ではほとんど学習が進まなくなる．$\theta = 0.0$ であれば，いつかは学習できるように思えるが，

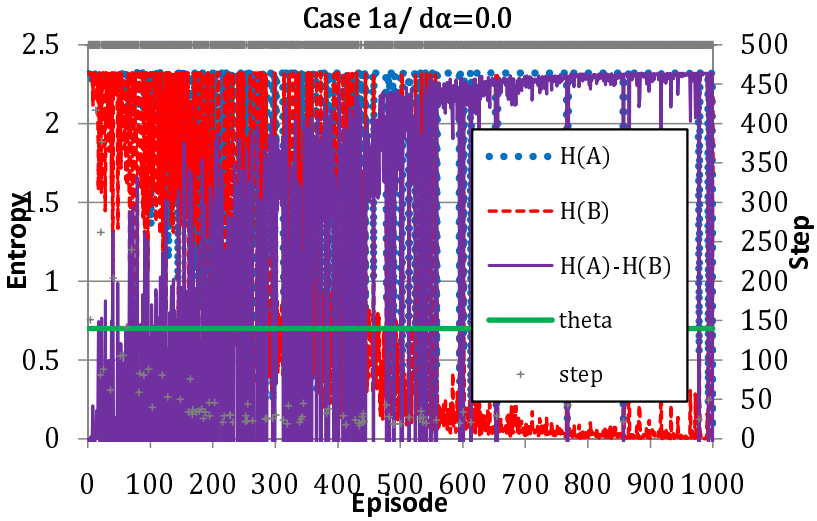



図 4.21　問題環境 b ケース 1a のエントロピー推移

ボルツマン選択はその温度 $T$ により次第に方策の偏りを強めていくことから，以前に推定していたことが後になり大きな影響を及ぼすようになるため，時間が長引くほど学習の成功の可能性がなくなってしまう．上記と同じ理由で，HighH エージェントの学習を早めるケース 2a では相手エージェントの方策が偏ること対して，価値推定したい状態が経験できないという状況を打破できない，また，序盤ではエージェント B の価値推定を早めてしまう効果さえ持っているといえるため，この二つの方法は競合回避に有効に働きにくいといえる．

**割引率 $\gamma$ 変更の効果**

最後に $\gamma$ 変更の効果について考察する．割引率 $\gamma$ の本来の意味は，報酬がすぐに得られるわけではない状態における行動の価値を将来得られる報酬からどれだけ割り引いて推定するかである．割引率を下げた場合，それまでより状態の価値を下げることになる．例えば，ある状態においてある行動が選択されると，それまで推定していた価値より低く更新される．さらに，状態-行動価値の更新は選択された行動に対して行うことから，価値が相対的に高い行動ほど，価値が下げられることになる．すると他の行動との価値の差が少なくなり，行動の偏りが減る，つまりエントロピーが上がる効果がある．この効果は価値が大きく偏っている状態-行動価値に対してより高い確率で発揮されることになるため，ケース 1a，2a のときのように急激にエージェント A の行動が確定的になった場合でも，その行動価値を重点的に下げる働きをすることで競合の発生を防ぐ．一方で HighH エージェントの $\gamma$ を上げるケース 2a のグラフにおいて特徴的なことは，学習の遅くれているエージェントはずっとエントロピーが高いままエピソードを継続し，あると

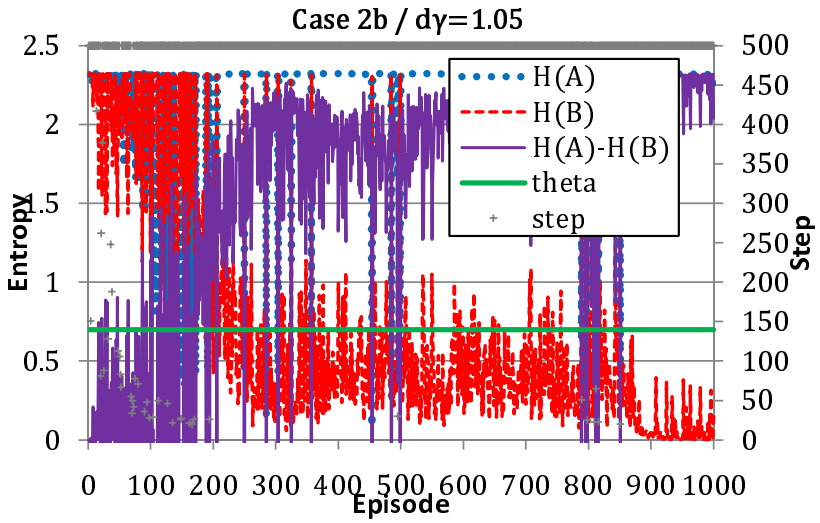



図 4.22　問題環境 b ケース 2a のエントロピー推移

きになって急激に学習を進めることであり，さらに，両エージェントが同時に目標に到達して MAX_EPISODE 以内に終わっているエピソード（つまり終了ステップ数が小さいエピソード）が非常に少ない点である．割引率を上げた場合，得られるであろう報酬を過大に評価することになる．この結果，その行動をより高い可能性で選択することにつながる．一方，負の報酬である通常報酬に対しても同様に過大に評価することで，常に得る負の報酬によりほとんどの行動が価値の低い（悪い）行動であると推定される．この事実は一定エピソード（MAX_EPISODE）を終えた時点での全ての $Q(s,a)$ をみることで確認されている．具体的に，図 4.20 に示された学習では，ほとんどの行動価値を低く推定したエージェント B が，一度ゴール報酬を得るとその行動価値を過大に評価することで一つの行動価値だけを非常に高く推定する．つまり，ゴール報酬につながる状態-行動価値だけが相対的に高く評価される．エピソードに基づくエントロピーではエピソードで経験した全ての状態のエントロピーの平均をとるため，ある状態における行動が確定的になっても他の多くの状態の非常に低いエントロピーの影響ですぐに下がることは少ない．そして目標状態の近くにある状態-行動価値から着実に推定を積み重ね，あるときに必要な行動が一気に確定的になる．このような影響は元々学習が進んでいるエージェントは受けにくい．なぜなら多くの状態の行動価値をまんべんなく低くすることは多くの時間を要し，偶然エントロピーの差が逆転したくらいでは時間的に不十分だからである．よって，状態-行動価値を過小評価した状態をつくりあげることが，以降に偶然得るゴール報酬から着実に価値推定することに有用であり，さらにこのようなエージェントの $\gamma$ を上げてコール報酬を過大に評価することが競合回避に有効であることが分かった．

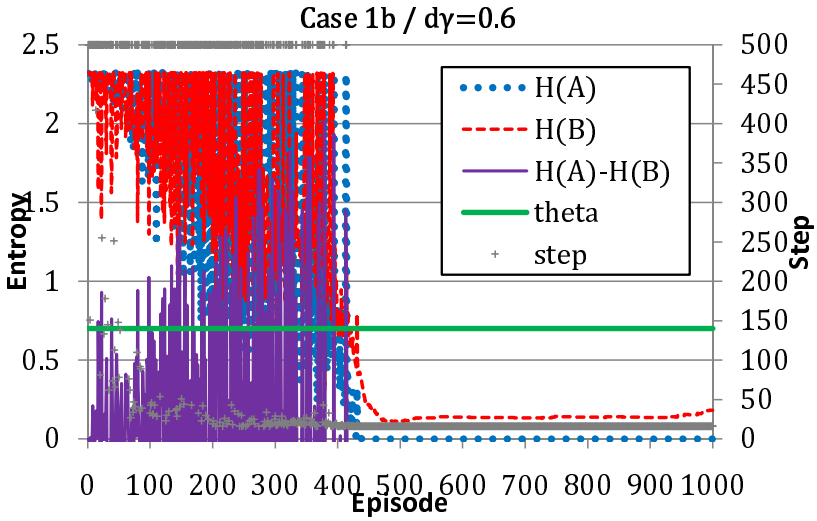



図 4.23　問題環境 b ケース 1b のエントロピー推移

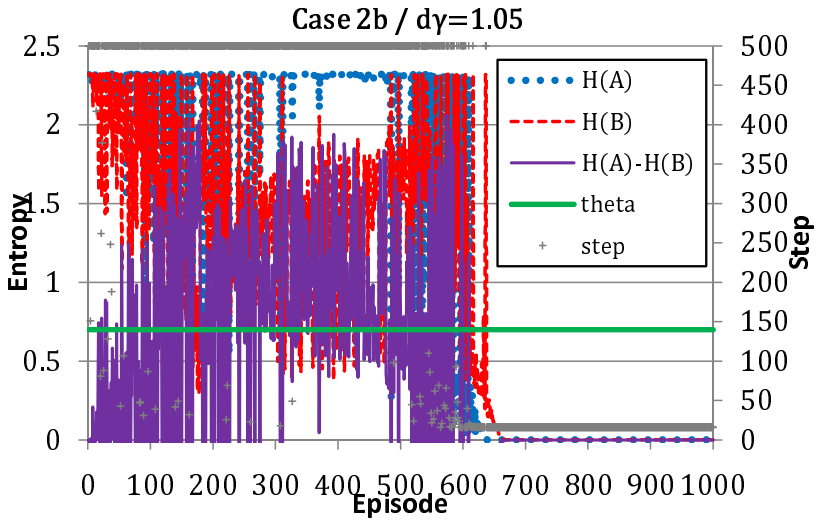



図 4.24　問題環境 b ケース 2b のエントロピー推移

## 4.5　実験２：三体エージェント

### 4.5.1　実験内容

三体エージェントに拡張した提案手法の有効性を検証するため，三体エージェントの狭路すれ違い問題（図 4.25）を新たに用意して計算機上に実装し，以下の表 4.4 に示すケースに別けて実験することで，それぞれの方法の効果の違いを明らかにする．縦軸はどんな

方針の手法であるか表しており，横軸はどちらの学習パラメータを変更するかを表している．

また，学習パラメータ変更の対象となるエージェントのエピソードに基づくエントロピーが偶然の行動選択によって逆転してしまい，本来学習パラメータを変更したいエージェントとは別のエージェントのパラメータが変更されてしまうことで，手法が効果的に働かない可能性が考えられる．そこでエピソードに基づくエントロピーの移動平均を利用しエピソード間のエントロピーのバラつきを抑えることで，手法がより有効となるかどうかを検証する．エントロピーの移動平均は過去 10 エピソードにおけるエントロピーの相加平均をとったものを扱う．移動平均エントロピーの検証も表 4.4 に示す八つのケースに別けて実験し，この場合ケースの後ろに“mv”と記載する．

表 4.4　実験２における実験ケース

| | 学習率 $\alpha$ を変更 | 割引率 $\gamma$ を変更 |
|---|---|---|
| LowerH エージェントの学習パラメータを下げる | ケース 1a-I | ケース 1b-I |
| LowestH エージェントの学習パラメータを下げる | ケース 1a-II | ケース 1b-II |
| HigherH エージェントの学習パラメータを上げる | ケース 2a-I | ケース 2b-I |
| HighestH エージェンの学習パラメータを上げる | ケース 2a-II | ケース 2b-II |

### 4.5.2　評価指標とパラメータ設定

評価は前章の実験と同様に評価する．新たに用意した問題環境 c のパラメータは以下のように設定する．

表 4.5　実験２の各問題環境とパラメータ

| 環境 | c（図 4.25） |
|---|---|
| エージェント数 | 3 |
| MAX STEP | 500 |
| MAX EPISODE | 5000 |

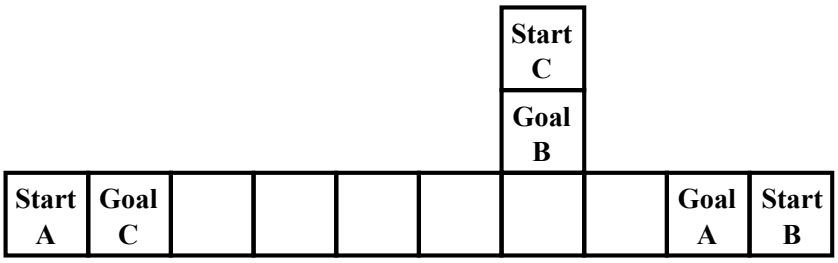



図 4.25　問題環境 c

### 4.5.3　実験結果

**ケース** 1a-I，1b-I，1a-II，1b-II

LowerH（少なくとも一体のエージェントより $\theta$ 以上低いエントロピーをもつ）エージェントの学習パラメータを下げる，ケース 1 の全ての結果を示す．図 4.26，4.27，4.28，4.29 はそれぞれのケースにおいて，$\theta$，$d\alpha$，$d\gamma$ を変更した実験の結果である．グラフは前章と同様のもので，縦軸は成功率，横軸は $d\alpha$ あるいは $d\gamma$，各ラインは同じ $\theta$ の結果を繋いだものを示している．結果をみると，ケース 1a の $\alpha$ を下げる手法についてはどちらの設定においても，通常の Q 学習の成功率 0% から変化せず効果がないことかわかる．LowerH，LowestH エージェントの割引率 $\gamma$ を下げることにより全く成功しなかった学習を成功させることができることがわかる．ただし全ての競合を回避するには至らない．$\theta = 2.0$ の場合と，それ以下の場合では $d\gamma$ が小さくなったときの傾向が異なり，前者は小さいほど成功率が上がり，後者はある範囲で成功率が上がるが，さらに小さくなると再び競合に陥る可能性が高くなる山型をとっている．また，ケース 1b の I と II で比較すると，山型の頂点の位置が異なる．それぞれ $\gamma$ を下げる対象が LowerH エージェント全てか，LowestH エージェントのみかであるため，学習パラメータ変更の総合的な量はケース 2 のほうが多い．そのため，ケース 2 では各エージェントが変更する大きさが少なくても効果が得られており，頻度が少ないケース 4 では一度に大きく変更する場合に高い効果が得らると考えられる．

次に，図 4.30，4.31，4.32，4.33 にエントロピーの移動平均を用いた結果を示す．結果をみると，ケース 1a の $\alpha$ を下げる手法についてはどちらの設定においても，移動平均を取ることが競合の回避に繋がらない結果となった．移動平均を取ることで変わったこととして，LowerH，LowestH エージェントの割引率 $\gamma$ を下げることにより全ての競合を回避するには至らないものの，移動平均を取らないエントロピーを利用した手法よりも高い成功率が得られていることがわかる．また，$\theta$ が小さい設定においても高い成功率が得られるようになったこととがわかる．特に，最良のパラメータは，最も大きい $\theta$=2.0 ではなく

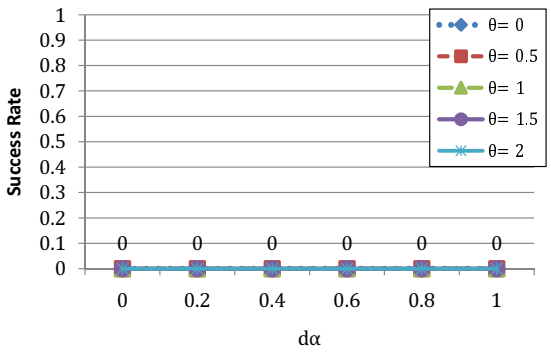



図 4.26　問題環境 c ケース 1a-I の成功率

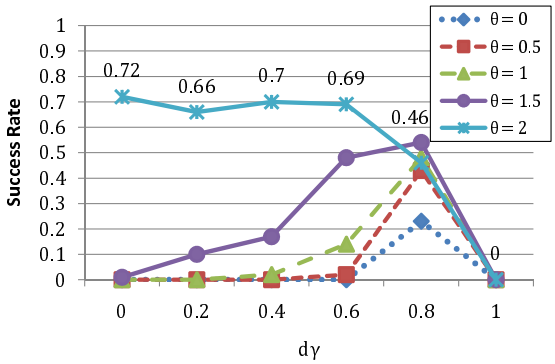



図 4.27　問題環境 c ケース 1b-I の成功率

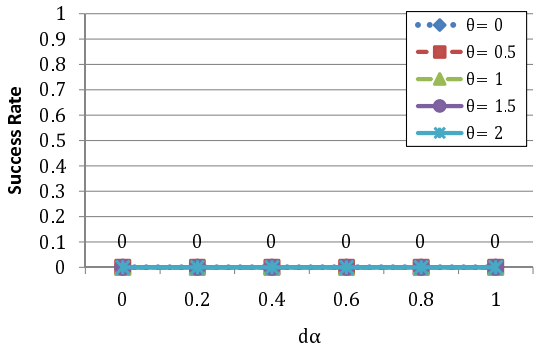



図 4.28　問題環境 c ケース 1a-II の成功率

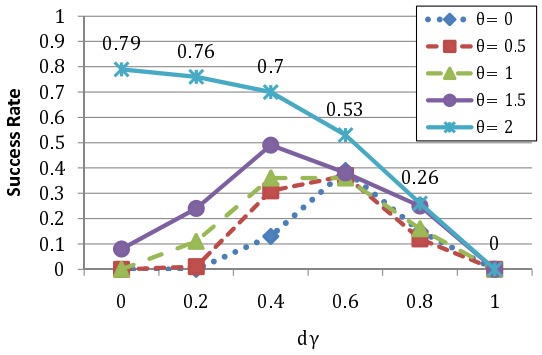



図 4.29　問題環境 c ケース 1b-II の成功率

$\theta$=1.5 となった．$\theta$ がさらに小さいとき，再び成功率が下がることから，一番成功率が上げられる設定は $\theta$=1.0 から 2.0 の間に存在していることが予想される．前の章で調べた学習の収束という観点からみると，これらのパラメータではまだ学習が収束していない可能性が大きい．この結果は，エージェントの行動が確定的（$\theta$=2.0）すぎず曖昧（$\theta$=1.0）すぎないような効率よく学習を進めるための丁度良い $\theta$ の値が存在していること示していると考えられる．

### ケース 2a-I，2b-I，2a-II，2b-II

次に，図 4.34，4.35，4.36，4.37 にケース 2 について，$\theta$，$d\gamma$ を変更した実験結果を示す．縦軸は成功率，横軸は $d\gamma$，各ラインは同じ $\theta$ の結果を繋いだものを示している．ケース 1a-I，1a-II はケース 2a-I，2a-II と同様に全く競合回避に効果がないことがわかる．一方で，HigherH，HighestH エージェントの割引率 $\gamma$ を上げるケース 2b-I，2b-II では共に $\theta = 0.0$ かつ $d\gamma = 1.05$ のとき成功率が最大となり，ほぼ全ての競合を回避できたといえる．この結果をみると $\gamma$ を単に上げればいいのではなく，$d\gamma = 1.04$ より大きく上げ

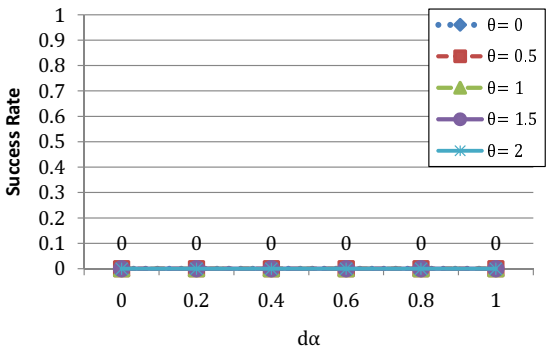



図 4.30　ケース 1a-Imv の成功率

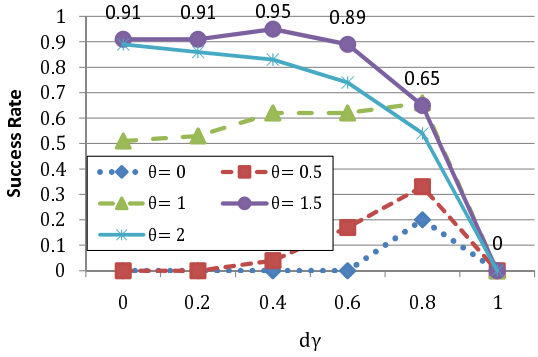



図 4.31　ケース 1b-Imv の成功率

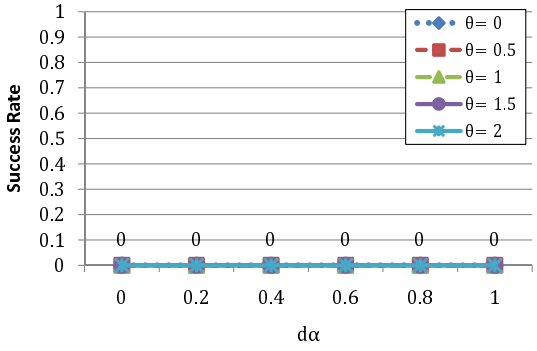



図 4.32　ケース 1a-IImv の成功率

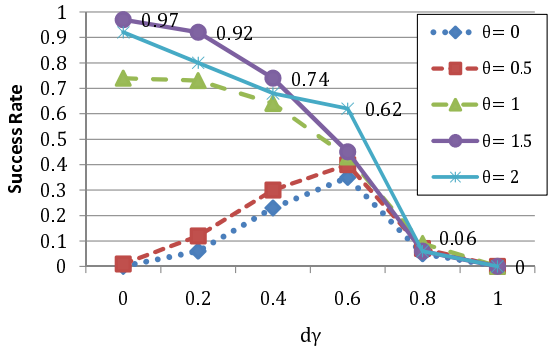



図 4.33　ケース 1b-IImv の成功率

るようにすることで，急激に競合回避に効果が現れることがわかる．また，ケース 2b-I と 2b-II を比較すると，後者は $\theta$ の大きさに依存して成功率が得られているように捉えられるが，前者はそのように整然とした結果が得られているわけではない．これは，行動が確定的になったエージェント二体に対して LowestH エージェントのみが学習パラメータを変更し学習することより，二体の LowerH エージェントが学習パラメータを変更して学習することが，より複雑な相互作用が生じる環境で学習しているといえるためと考えられる．

次に，図 4.38，4.39，4.40，4.41 にケース 2 について，$\theta$，$d\gamma$ を変更した実験結果を示す．縦軸は成功率，横軸は $d\gamma$，各ラインは同じ $\theta$ の結果を繋いだものを示している．やはり $\alpha$ を上げる手法では競合回避の効果を得られないことがわかる．一方で，HigherH，HighestH エージェントの割引率 $\gamma$ を上げるケース 2b-I，2b-II では共に $\theta = 0.0$ かつ $d\gamma = 1.05$ のとき成功率が最大となり，ほぼ全ての競合を回避できたといえる．ケース 2 の結果はケース 1 の結果と比べて移動平均を取ることによる変化が小さい．なぜなら，学習が遅れているエージェントの $\gamma$ を上げても，負の報酬に対して過小評価をする以外には

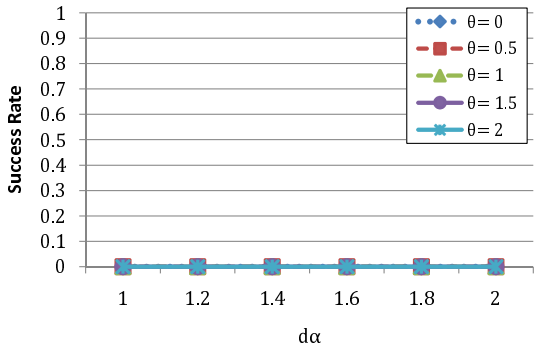



図 4.34　問題環境 c ケース 2a-I の成功率

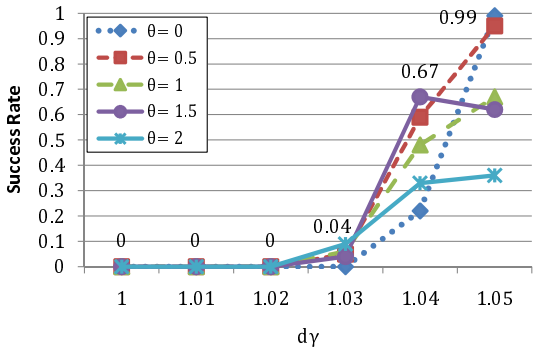



図 4.35　問題環境 c ケース 2b-I の成功率

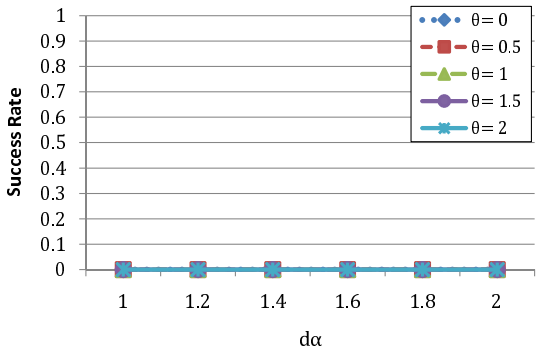



図 4.36　問題環境 c ケース 2a-II の成功率

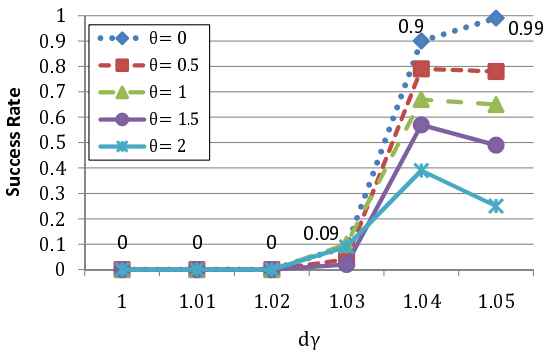



図 4.37　問題環境 c ケース 2b-II の成功率

すぐにはエントロピーの大きさを変えるようなことが起こらないからである．また，エントロピーが下がるような効果が現れるとしても，エピソード中のほぼ全ての状態において行動が曖昧でない学習が進んでいるエージェントにはなんら変更が加えられてないため，エントロピーの上下関係が変わることは容易にはない．その為，エピソード間でのエントロピーの揺らぎを減らすためである移動平均を取ることの影響はほとんどないと言える．

### 4.5.4　考察

ここでは，エージェント数が三体に増えたことで，提案手法の効果にどういった違いが生まれたのか，また問題のクラスがどのように変わったかを分析することで，エージェント数が n のときの有効な手法を探求する．$\gamma$ を変更することによる基本的な効果の考察は前章の考察を参照されたい．

この分析のために，前章と同様に，環境 c のそれぞれのケースにおけるエージェントのエントロピーの変化を全ての乱数シードの結果の平均をプロットしたグラフを図 4.17〜

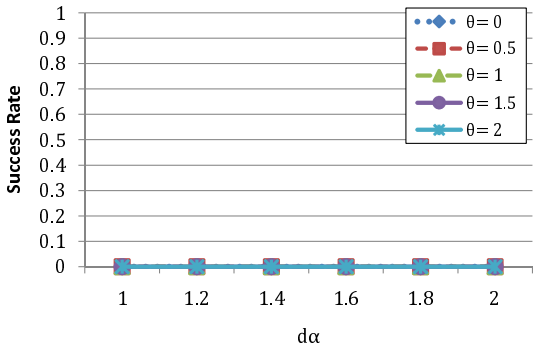



図 4.38　ケース 2a-Imv の成功率

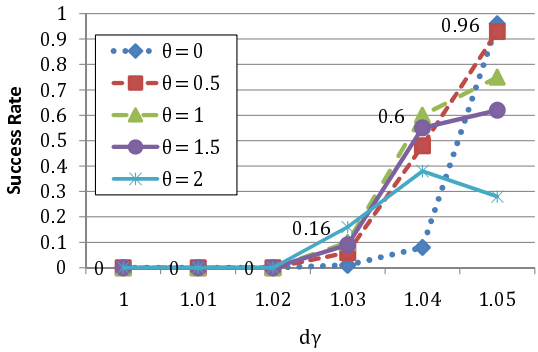



図 4.39　ケース 2b-Imv の成功率

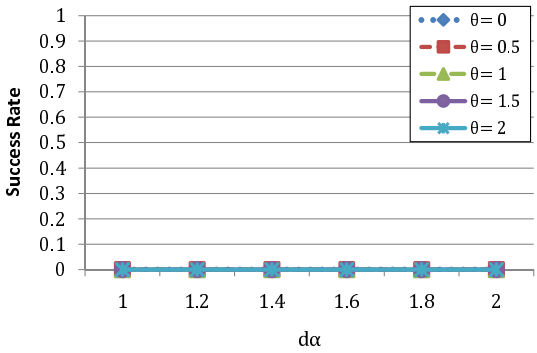



図 4.40　ケース 2a-IImv の成功率

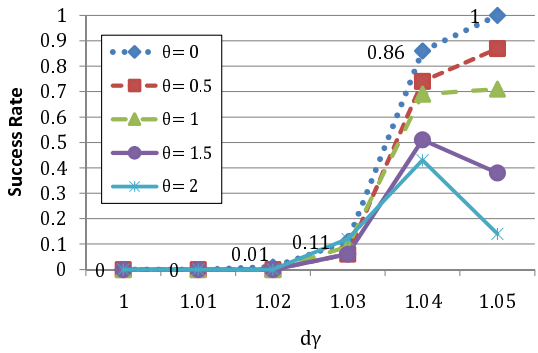



図 4.41　ケース 2b-IImv の成功率

4.20 に示す．これらの設定は全てケース毎で最良の結果をもたらしたもの，つまり，ケース 1b − I，II では $\theta = 2.0$，$d\gamma = 0.0$ ケース 2b − I，II では $\theta = 0.0$，$d\gamma = 1.05$ という設定をケースの代表とした．これらグラフ（特にケース 1b）をみると，エージェントの学習が成功するためには複雑な学習過程を要することがわかる．具体的には，まずゴールが近いエージェントから順に学習を進め，手法の効果によってエージェント B あるいは A が学習が滞る間に C が第一段階目の学習（これはおそらくエージェント B の方策に対して報酬をより多く得るような学習）を進めた後，エージェント C のある程度偏った方策に依存したなんらかの学習を A が進めるという段階を経て，最終的に全エージェントがゴール報酬を得るような状態-行動価値をみつけられたと推察できる．そのような段階的な学習は言い換えれば，複数の競合が介在しており，それらを全て回避しなければ，全体としての競合回避が不可能であるといえる．

次に，ステップ数をみると $\gamma$ を下げるケース 1b はまだ学習途中の段階であり，まだ少し先をみれば成功率が上がると考えられる．一方，HigherH，あるいは HighestH エージェントの $\gamma$ を上げるケース 2b については，ステップ数は横ばいになりこれ以上は上が

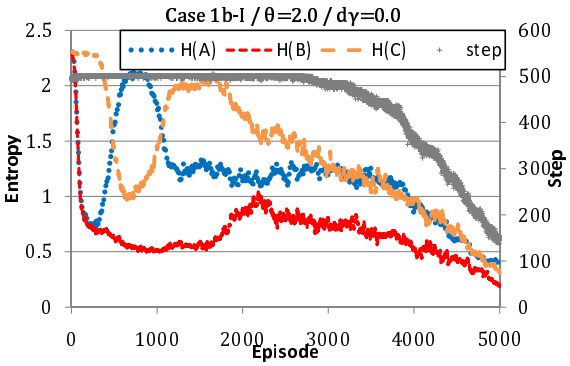



図 4.42　問題環境 c ケース 1b-I のエントロピー推移の平均

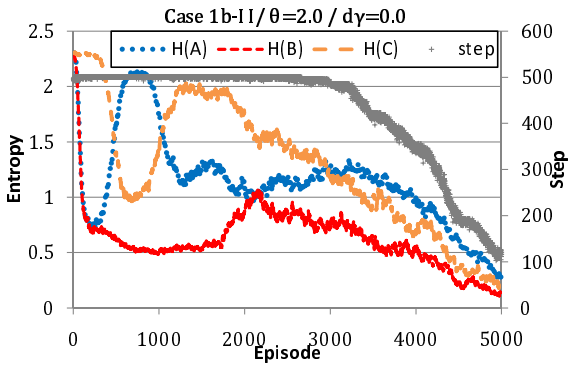



図 4.43　問題環境 c ケース 1b-II のエントロピー推移の平均

ることはないと考えられる．これより，成功率が 100% となっていないことを考えると，時間がかかってもケース 1b のような手法での学習の収束を待つことが場合により有益であると考えられる．ケース 1b と 2b の大きな違いは，前者では最も学習が進んでいるエージェントのエントロピーを上げて他のエージェントの目標到達確率を増やす方法である一方，後者はその確率はそのままに，数少ない目標達成時における推定を効率よくする方法であることである．一方で共通することは，最も学習が進んでいるエージェントの行動が学習の結果に大きく影響することである．具体的には，ケース 1b については前述の通りであり，ケース 2b ではその二つの方法の収束速度の差の原因が最もエントロピーが低いエージェントのエントロピーの違いにあると考えられる．この結果においては，全ての HigherH を $\gamma$ 変更の対象とする I では最も低いエントロピーを持つエージェントの行動が，他方 II のそれよりも確定的でないこという差異が，他エージェントの目標到達確率を上げ，学習の収束を早くする原因であるといえる．そして，これを引き出したのは他でもなく $\gamma$ 変更の対象の違いに由来し，つまり対象が LowestH エージェントのみであるケース 1b-II においてはエントロピーの高いエージェントがたまたまエントロピーを下げても，同時期にもう一方のエージェントがエントロピーを下げることは稀である一方で，対象が LowerH エージェントであれば，エージェント B に対しても $\gamma$ の変更が適用される可能性が増えることでエントロピーを上げることにつながると考えられる．

またここで，学習が進んでいるエージェントの割引率を下げる手法は，状態-行動価値を下げ，エージェントの行動を曖昧にすることで，他エージェントの目標達成の可能性を上げることで，学習を促進し競合を解消したと言えることから，その手法と初めから行動確率に一定の割合でランダム選択が入る行動選択手法を用いた Q 学習の成功率で比較することで，提案手法が単純にランダムがある方策よりも優れていることを確認する．ここで比較として用いる行動選択手法は $\epsilon$-greedy [23] と呼ばれる手法である．これは 0 から 1 までの小さい値に設定した $\epsilon$ 値と同じ確率でランダム行動を取り，そうでないとき（確

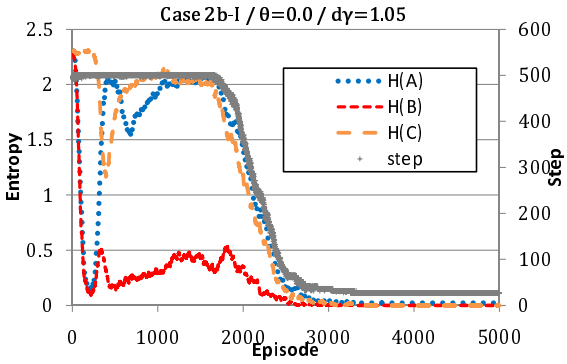



図 4.44 問題環境 c ケース 2b-I のエントロピー推移の平均

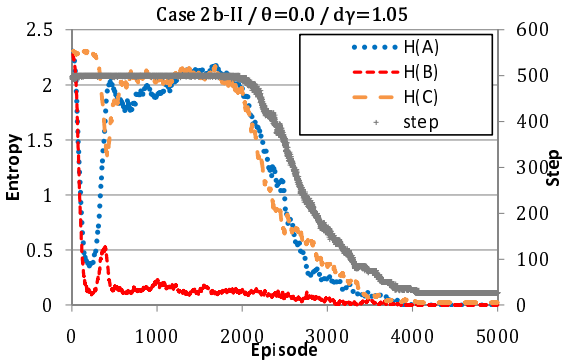



図 4.45 問題環境 c ケース 2b-II のエントロピー推移の平均

率 1-$\epsilon$）は最も高い価値の行動を選択する手法である．このように必ずランダムの選択が入るため，学習した行動価値を選択に反映しながらも，様々な状態を探索することで局所最適解に陥りにくい，単純だが優れた選択手法である．

実験環境 c に $\epsilon$-greedy 手法（$\epsilon$=0.05）を用いた Q 学習エージェントを適用し，100 個の乱数シードにおいて 5000 エピソード学習した結果，学習が「成功」した確率（成功率）は 72% であった．ボルツマン選択による Q 学習では全く成功しなかったことに対して，高い成功率を示すという結果を得た．競合回避の能力で言えば，提案手法の数ケースほど競合が解消できないといえる．

次に学習の成功率のエピソード毎の推移を図 4.46 に $\epsilon$-greedy と実験 2 において学習が進んでいるエージェントの割引率を下げる手法についての結果を示す．この結果をみると，成功率の立ち上がりは $\epsilon$-greedy が最も早いことが明らかであり，次に移動平均エントロピーを用いた提案手法が早いことが分かった．移動平均を用いない手法は結果的には $\epsilon$-greedy と同程度の成功率を示したが，5000 エピソード付近では，$\epsilon$-greedy 手法は成功率がほぼ横ばいになっている一方で，提案手法はまだ角度があるためエピソードを長くすれば，成功率がさらに上げられることが期待できることが明らかである．$\epsilon$-greedy 手法でさらに成功率を上げるには $\epsilon$ 値をさらに高くし，ランダム性を上げる必要があると考えられるが，その場合は学習速度は遅くなると考えられる．また，ランダム性を上げることは，上手く目標座標に到達する確率を減らし，いつまでも学習が進まなくなってしまう可能性を生むことも注意しなければならない．

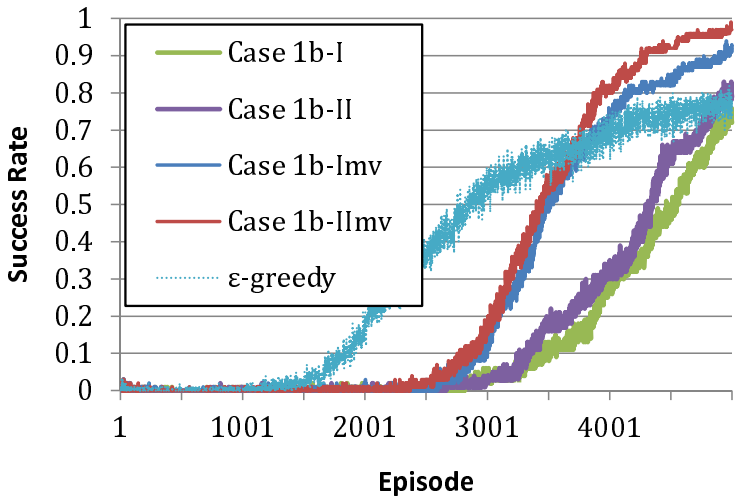



図 4.46　ケース 1b-I，II，1b-Imv，IImv と $\epsilon$-greedy の成功率推移

## 4.6　知見

### 4.6.1　競合回避の可能性

数種類の競合問題に対して提案手法を適用したシミュレーションの結果，大きく分けて二つの手法が競合回避に有効であることがわかった．(1) 学習が進んでいるエージェントの割引率 $\gamma$ を下げる方法は，価値が高く選択されやすい行動価値を重点的に下げ，方策の偏りを防ぐことで，全てのエージェントが目標達成のために学習できる機会を増加させる働きがあり，(2) 学習が遅れているエージェントの割引率 $\gamma$ を上げる方法は，目標達成につながる行動とつながらない行動の価値をはっきり分けるように推定する性質をもち，ゴール報酬獲得の少ない機会から非常に効率よく学習する働きがあることが明らかとなった．

### 4.6.2　学習進度の有用性

上記の競合解消はエージェント間の学習進度の差という相対的な指標がなくしては成り立たない．なぜなら，(1) 全てのエージェントの割引率 $\gamma$ を一様に下げることは，単に全てのエージェントの学習を停滞させるだけであるし，(2) 得られたゴール報酬が他エージェントと競合しない行動に対するものかどうかは，一方の行動がある程度確定的であることが大きな保障であるからである．つまり，単に（いつでも誰に対しても）学習パラメータを変更することが良いわけではないのであり，それに対して学習進度という情報

は，競合の原因となる方策の偏りを的確に表していることから，学習進度の有用性は非常に高いといえる．

### 4.6.3 定量化の的確さ

学習進度の定量化手法として本論文で提案したエピソードに基づくエントロピーは全ての状態-行動価値に基づくエントロピーと比較して，(i) 現在の方策の偏りをより直接的に表すこと，(ii) 計算コストが低いことの二点に優れている．具体的には，(i) は持っている全ての状態-行動価値と現在よく用いられる状態-行動価値が異なることに追従することが，方策の偏りを的確に表せる方法であり，(ii) は計算に用いる状態数に依存することから，全ての状態-行動価値に基づくエントロピーでは観測しうる状態数であり，提案手法では最大でも MAX_STEP の数であるから，その値は状態数を超えて設定することはほとんどない．エピソードに基づくエントロピーによる学習進度の評価を用いると，競合問題のように極端に学習が全くできない（ゴール報酬などの高い報酬が得られない）エピソードとそうでないエピソードで，エントロピーの値が激変する．現状では，ゴール報酬が得られなかったエピソードを区別できるほど差が現れることが，競合回避に有利に利用されている可能性が考えられ，公平でないことが懸念される．この問題を解決するために，移動平均やその他の平均化法を適用することが考えられるが，単純に適用した場合，それに付随する問題を個別に解決する必要がある．例えば，移動平均では，ずっと中間の値をとっているのか，それとも高い値と低い値を同等の確率でとっているのか区別がつかない．また，行動の結果が同じ行動の価値が等しく高く評価された場合，結果として行動に偏りが生じるはずであるが，この定量化においては，等しく高い行動選択確率からエントロピーを高く計算してしまう．これは定量化を的確にするために解決しなければならない問題である．

### 4.6.4 提案手法の有効性

提案手法はシミュレーション上で競合回避に対して有効であることを示した．通信量に関しては，行動選択回数の数百分の一の頻度であり，共有する情報は唯一つの実数値であり，非常に少ないといえる．しかし，設定値への依存度が高く，シミュレーションではパラメータを網羅的に設定しただけで，自動的に導き出すような機構は組み込まれていないため，実用を考えるとまだ検証不足である．一方で，多数の手法を細かく分析した結果，設定値についても有益な知見が得られている．ここで提案手法の実用を想定すると手法とパラメータを，(a)HigherH エージェントの割引率を上げる手法（$\theta = 0.0/d\alpha = 1.0/d\gamma \approx \frac{\gamma_{max}}{\gamma_0}$），(b)LowestH エージェントの割引率を下げる手法（$\theta \approx H_{max}/d\alpha = 1.0/d\gamma = 0.0$），のいずれかにすることが有効であることを挙げる．し

かし，これに関しては問題依存性が高く感じるため，さらに深く調査しなければならない．

# 第 5 章

# 内部報酬に基づく大域的最適解探索

マルチエージェント学習ではタスク割り当ての観点を含めて学習させるように環境設計されることは少なくない．そういった環境のモデルにおいても，従来のエージェントは外部から与えられる報酬をそのままの報酬値として利用するが，目標状態が複数ある問題では容易に局所政策の獲得に陥る要因になり得る．与えられた報酬値をそのままの大きさとして受け取るのではなく，エージェントが置かれた状況によって報酬値を独自に解釈し直すことによって，大局的な最適政策の獲得を可能にするエージェントを構築する．具体的には，「満足度」のような概念を考慮し，ある時間までに得た報酬の記憶から相対的に満足できる報酬値を求め，満足度から外部報酬に対する適正な報酬（内部報酬）を算出することで，高い報酬が獲得できる目標状態を探索する．これはマルチエージェント強化学習のモデル全体から見ると，図 5.1 の黄色の部分のモデルが対象である．

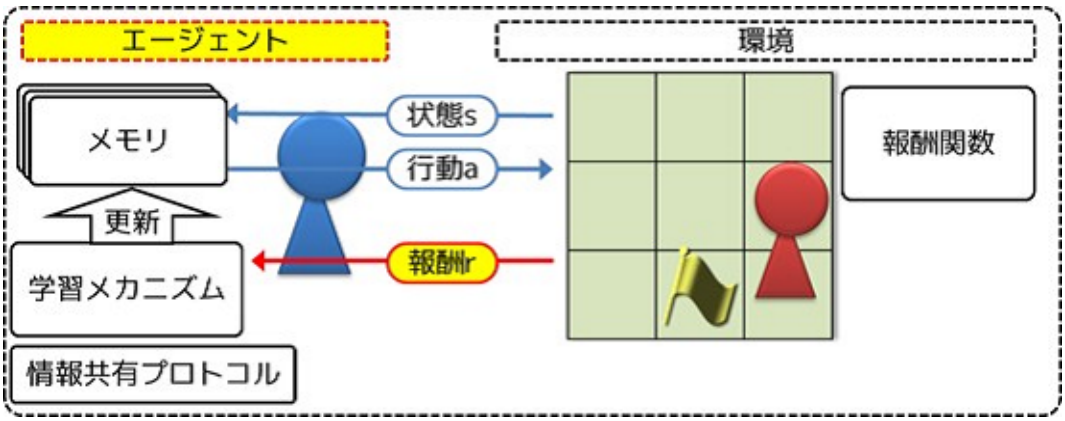



図 5.1 本章が対象とする領域

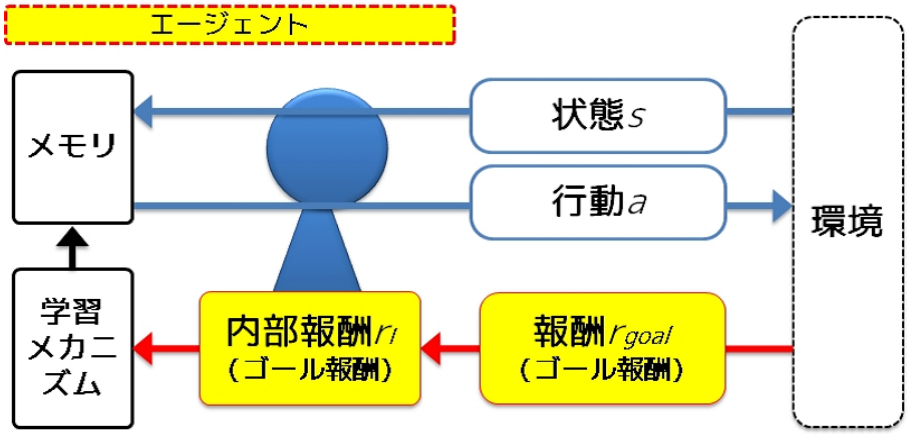



図 5.2　獲得した外部報酬を内部報酬へ変換して学習に用いるエージェント

## 5.1　内部報酬に基づくエージェント

### 5.1.1　概要

本研究では，局初解に陥らず最適解を見出すために，実際に獲得するゴール報酬とは別に Q 値の更新のための内部報酬 $r_I$ を導入したエージェント 1 ～ 3 を提案する．提案する三つのエージェントは，図 5.2 に示すように，環境から与えられる報酬をそのまま用いるのではなくエージェント内部で変換した値（内部報酬値）を用いる．具体的には，経過学習時間（*episode*）と平均獲得報酬（$r_{ave}$）という二つの観点から導出される．詳しくは後述する．なお，$r_I$ はゴール報酬獲得時のみ式 (2.1) 内の $r$（実際は $r_{goal-near}$ あるいは $r_{goal-far}$）に置き換えて Q 値の更新に用いる．なお以後の $r_{goal}$ は，該当するエピソードで得たゴール報酬の値（$r_{goal-near}$ あるいは $r_{goal-far}$）である．

### 5.1.2　アーキテクチャ

本章の提案エージェントの機能も基本エージェントがベースである．エージェントは通信機能を有さず，他エージェントと情報のやりとりは行わずに完全に独立して学習を行う．基本エージェントとの差分として，記憶メモリと計算量の増加がある．具体的には，内部報酬値の計算にかかる計算量と，エージェント 2 と 3 では現在の報酬の平均値 $r_{ave}$ を保持するメモリが必要がある．

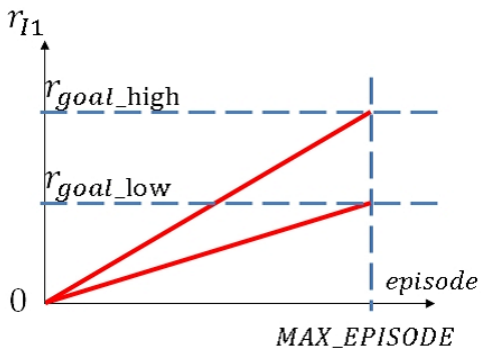



図 5.3　エージェント 1 の内部報酬

### 5.1.3　メカニズム

**エージェント 1 ：時限増加内部報酬**

時限増加内部報酬では図 5.3 に示すように時間の経過によって初めは小さく，徐々に大きくなるように内部報酬を見積もる．そうすることで学習初期に相対的に報酬の高いゴールへ向かう行動の偏りを低減させることができるため多くの状態の探索が期待できる．これによって偏ったゴール方向への過学習が抑制されて局所解だけでなく，最適解を見つける可能性を広げる．具体的に，内部報酬 $r_{I1}$ は下記の式で算出される．

$$r_{I1} = r_{goal} \times f \tag{5.1}$$

$$f = episode/\mathrm{MAX\_EPISODE} \tag{5.2}$$

ここで，$f$ は最大エピソード（MAX_EPISODE）のとき最大値 1 をとるような時間（$episode$）の一次関数である．

**エージェント 2 ：平均差分内部報酬**

平均差分内部報酬では，現エピソードで得たゴール報酬から今までに得たゴール報酬の平均値を差し引いたものを内部報酬とする．これによって平均より小さいゴール報酬（局所解）方向へは近づかず別の大きい報酬（最適解）を求める探索が促進されることが期待できる．具体的に，内部報酬 $r_{I2}$ は下記の式で算出される．

$$r_{I2} = r_{goal} - r_{ave} \tag{5.3}$$

ここで $r_{ave}$ は今までに得たゴール報酬の平均である．図 5.4 に示すように，例えば報酬が二値である場合は獲得するゴール報酬の平均値は二つの間の値をとるため，相対的に低い報酬が与えられるゴール状態から算出される内部報酬は負の値となり，エージェントは

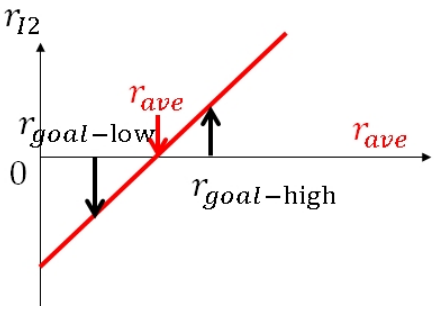



図 5.4　エージェント 2 の内部報酬

その報酬へ向かおうとしなくなる．報酬が二値でなくても，徐々に平均報酬が増加することが期待できるため，無限時間の学習では平均報酬値が最大のゴール報酬値に漸近することが期待できる．実際には無限時間の学習を行わないため，算出される内部報酬はゼロより少し大きい値をとるようになると考えられる．

**エージェント 3 ：平均差分＋時限増加内部報酬**

第三の提案エージェントでは平均差分内部報酬で用いる平均値の影響を時間の経過によって徐々に小さくするように内部報酬を見積もる．上記二つの手法を組み合わせた手法で，具体的に，内部報酬 $r_{I3}$ は下記の式で算出される．

$$r_{I3} = r_{goal} - (1 - f) \times r_{ave} \tag{5.4}$$

ここで，$f$ は式 (5.2) で表され，最大エピソード（MAX_EPISODE）のとき最大値 1 をとるような時間（episode）の一次関数とする．したがって，図 5.5 に示すように学習の初めはエージェント 2 と同様の内部報酬を算出するが，最終的に $(1 - f)$ は 0 に近づくため，式中の $r_{ave}$ の項はなくなり，内部報酬は環境から与えられる報酬と同等となる．また，エージェント 2 と比較して今までに得たゴール報酬の平均値が徐々に小さくなるため，ゴール方向への Q 値を負の値に見積もる可能性が低減する．

**アルゴリズム**

最後に内部報酬に基づくエージェントのアルゴリズムを **Algorithm 2**(1)〜(3) として示す．大文字斜体で書かれた単語は関数，その他の大文字は定数を表しており，$Env$ は環境の略称でありドットに続く情報はエージェントではなく環境が保持している．ID はエージェントの識別符号を表している．

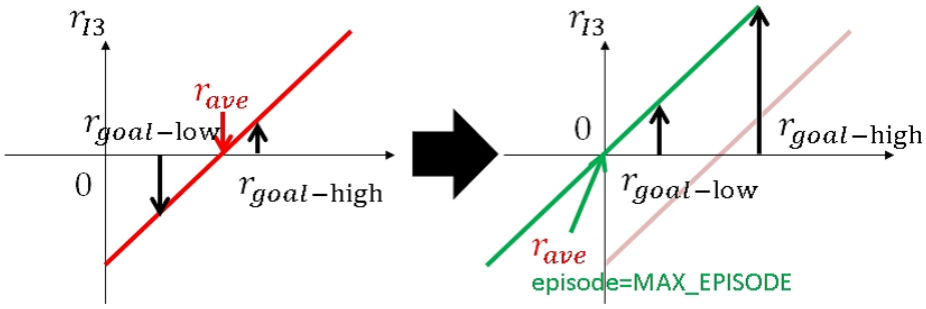



図 5.5　エージェント 3 の内部報酬

**Algorithm 2(1)**　Q-learning based on internal time-control reward

$\alpha \leftarrow \alpha_0 \quad \gamma \leftarrow \gamma_0$
$episode \leftarrow 0$
**while** $episode <$ `MAX_EPISODE` **do**
　$s \leftarrow Env.StartState(\mathrm{ID})$
　$step \leftarrow 0$
　**while** $step <$ `MAX_STEP` $ors \not\equiv Env.GoalState(\mathrm{ID})$ **do**
　　$StoreState(s)$
　　$a \leftarrow SelectAction(Q(\mathcal{S}, \mathcal{A}))$
　　$DoAction(a)$
　　$s' \leftarrow Env.State(\mathrm{ID})$
　　$r \leftarrow Env.Reward(s')$
　　$r_I \leftarrow CalculateRewardI(r, episode,$ `MAX_EPISODE`$)$
　　$Q(s,a) \leftarrow UpdateQ(r_I, s', A', \alpha, \gamma)$
　　$step \leftarrow step + 1$
　　$s \leftarrow s'$
　**end while**
**end while**

**Algorithm 2(2)** Q-learning based on internal differential average reward

$\alpha \leftarrow \alpha_0 \quad \gamma \leftarrow \gamma_0$
$episode \leftarrow 0$
$r_{ave} \leftarrow 0$
**while** $episode <$ `MAX_EPISODE` **do**
  $s \leftarrow Env.StartState(\mathrm{ID})$
  $step \leftarrow 0$
  **while** $step <$ `MAX_STEP` $ors \not\equiv Env.GoalState(\mathrm{ID})$ **do**
    $StoreState(s)$
    $a \leftarrow SelectAction(Q(\mathcal{S}, \mathcal{A}))$
    $DoAction(a)$
    $s' \leftarrow Env.State(\mathrm{ID})$
    $r \leftarrow Env.Reward(s')$
    $r_I \leftarrow CalculateRewardI(r, r_{ave})$
    $Q(s, a) \leftarrow UpdateQ(r_I, s', A', \alpha, \gamma)$
    $step \leftarrow step + 1$
    $s \leftarrow s'$
  **end while**
  $r_{ave} \leftarrow CalcurateAverage(r)$
  $StoreAverage(r_{ave})$
**end while**

**Algorithm 2(3)**

Q-learning based on internal time-control differential average reward

$\alpha \leftarrow \alpha_0 \quad \gamma \leftarrow \gamma_0$

$episode \leftarrow 0$

$r_{ave} \leftarrow 0$

**while** $episode <$ `MAX_EPISODE` **do**

  $s \leftarrow Env.StartState(\mathrm{ID})$

  $step \leftarrow 0$

  **while** $step <$ `MAX_STEP` $ors \not\equiv Env.GoalState(\mathrm{ID})$ **do**

    $StoreState(s)$

    $a \leftarrow SelectAction(Q(\mathcal{S}, \mathcal{A}))$

    $DoAction(a)$

    $s' \leftarrow Env.State(\mathrm{ID})$

    $r \leftarrow Env.Reward(s')$

    $r_I \leftarrow CalculateRewardI(r, r_{ave}, episode,$ `MAX_EPISODE`$)$

    $Q(s, a) \leftarrow UpdateQ(r_I, s', A', \alpha, \gamma)$

    $step \leftarrow step + 1$

    $s \leftarrow s'$

  **end while**

  $r_{ave} \leftarrow CalcurateAverage(r)$

  $StoreAverage(r_{ave})$

**end while**

## 5.2 実験 1

### 5.2.1 実験内容

内部報酬に基づくエージェントの有効性を分析するため，3.2 章で示したマルチステップ複数報酬問題を計算機上に実装し，表 5.1 に示すケースに分けて実験した．ケース 1，2 は比較手法，3〜5 は提案手法である．ケース 2 のエージェントは他のエージェントと違い温度定数を $T = 1.0$ で一定とし，行動選択のランダム性を高く保つことでより探索を促せるような設定である．

表 5.1　実験ケース

| ケース | エージェント |
|---|---|
| ケース 1 | Q 学習エージェント |
| ケース 2 | Q 学習エージェント ($T = 1.0$) |
| ケース 3 | 提案エージェント 1（時限増加内部報酬エージェント） |
| ケース 4 | 提案エージェント 2（平均差分内部報酬エージェント） |
| ケース 5 | 提案エージェント 3（時間増加＋平均差分内部報酬エージェント） |

### 5.2.2 評価指標とパラメータ設定

実験は各ケースでシードを変えて 100 試行を行い，最終的にエージェントが獲得した政策を評価した．具体的には，$MAX\_EPISODE$ 回目のエピソードで行動選択手法をグリーディ選択手法（greedy selection）に変更し，つまり各状態で $Q(s, a)$ が一番大きい最良の行動のみを選択させたときに，エージェントが獲得した報酬と，報酬獲得までのステップ数を評価する．最良の結果はエージェントが 100 回の試行全てにおいて高報酬を最短ステップで獲得することである．具体的には，双方のエージェントが報酬 10 を 5 ステップで獲得することである．表 5.2 に実験パラメータを示す．$MAX\_STEP$ はエージェントがなかなか目標状態に辿り着かない場合に学習を打ち切るステップ数である．

### 5.2.3 実験結果

図 5.6 に各実験ケースの結果を示す．この結果はヒストグラムであり，縦軸は 100 試行中の頻度を表し，横軸は報酬を獲得したステップ数を表しているが，正の値にはエージェントが高報酬である 10 を獲得した場合，負の軸にはエージェントが低報酬である 5 を獲

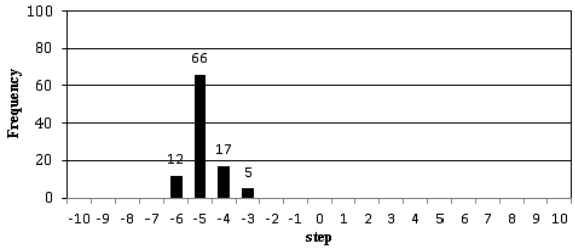
表 5.2 実験パラメータ

| | |
|---|---|
| $MAX\_STEP$ | 100 |
| $MAX\_EPISODE$ | 1000 |
| $\alpha_0$ | 0.2 |
| $\gamma_0$ | 0.9 |

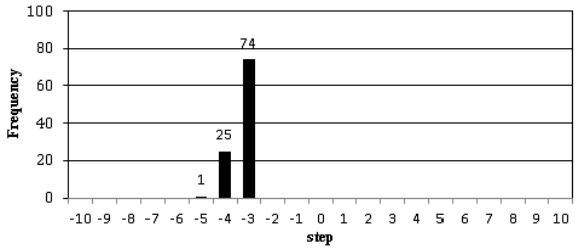
得した場合を表している．$MAX\_EPISODE$ において $MAX\_STEP$ 行動しても報酬を獲得できない場合（なんらかの理由で学習が上手くいかなかったことを示している場合）はカウントしない．前述した通り，最良の結果は毎試行で高報酬を最短ステップで獲得することであるため，このお結果においては +5 の頻度が 100 となることである．

以下で結果について議論する．

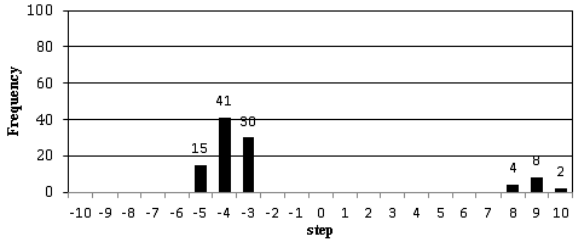
- ケース 1 : Q 学習エージェント
  図 5.6(1) は通常の Q 学習エージェントが局所的な政策に収束し，高報酬を獲得できなかったことを示している．
- ケース 2 : Q 学習エージェント ($T = 1.0$)
  図 5.6(2) は $T = 1.0$ から変化させない Q 学習エージェントもケース 1 と同様に局所的な政策に収束して高報酬を獲得できなかったことを示している．さらに，ケース 1 よりも行動選択のランダム性が高いため，より短い経路を見つけたことがわかる．
- ケース 3 : 提案エージェント 1 (時限増加解部報酬エージェント)
  図 5.6(3) は正のステップ数を示していることから，エージェントが高報酬を獲得できたことがわかる．
- ケース 4 : 提案エージェント 2 (報酬差分内部報酬エージェント)
  図 5.6(4) は平均差分内部報酬によってエージェントが局所的な政策への収束の回避に成功し，高報酬を獲得したことを示している．いくつかの試行において最小ステップで報酬を獲得できていることも見られる．一方で，他のいくつかの試行で報酬を獲得できない場合があり，頻度の合計は 100 未満であった ($66 = 28 + 35 + 2 + 1 < 100$).
- ケース 5 : 提案エージェント 3 (時限増加＋平均差分内部報酬エージェント)
  図 5.6(5) は平均報酬 $r_{ave}$ の影響を時間とともに小さくすることでエージェントが局所的な政策への回避に成功し，さらに，ケース 4 のように学習の失敗する試行がなくなったことを示している．その一方で，最短ステップでは報酬を獲得できなかったことがわかる．



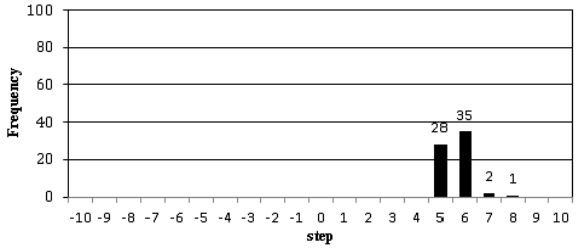

(1) ケース 1 : Q 学習エージェント



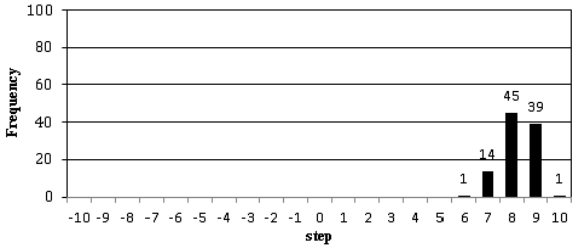

(2) ケース 2 : Q 学習エージェント ($T = 1.0$)




(3) ケース 3 : 提案エージェント 1 （時間増加内部報酬エージェント）




(4) ケース 4 : 提案エージェント 2 （平均差分内部報酬エージェント）




(5) ケース 5 : 提案エージェント 3 （時間増加＋平均差分内部報酬エージェント）

図 5.6　獲得報酬と獲得ステップ数のヒストグラム

### 5.2.4 考察

**行動のランダム性の影響**

ケース 1 とケース 2 の結果より，通常の Q 学習エージェントよりも温度 $T$ を高く保つエージェントの方が短いステップで報酬を獲得出来ることが明らかになった．これはシングルエージェント強化学習の場合でも自明のことで，推定した Q 値の情報を用いてそれに従って確定的に行動する（exploitation）か，ランダム性を高めて周辺を探索する（exploration）かのバランスを取ることは重要な問題である．この場合，より早く確実に報酬へたどり着く政策を獲得できたのは前者で，後者は周辺の探索を続けることによって短い経路を見つけることができたといえる．ランダム性の高め方によっては，あらゆるの政策の中から最も優れた政策を見出すことを可能にするが，本実験の設定の範囲ではそれを確認することができなかった．このランダム性に関する問題は学習に試行錯誤を伴う Q 学習ではいつでも留意しなければならない問題である．一方で，複数報酬の問題という視点で見ると，この 2 ケースに関してはいわゆる局所解に対してどれだけ収束しているかを議論しているに過ぎず，ランダム性に頼らない学習方法が求められる．

**内部報酬の利用の影響**

ケース 3〜5 の結果より，ゴール時の Q 値の更新において内部的に求めた報酬を用いることが複数ある報酬の内の高い報酬を獲得出来る政策を見つけることが出来ることが明らかになった．以下で，各内部報酬が与える影響について議論する．

- ケース 3 : 提案エージェント 1 (時限増加解部報酬エージェント)
  ケース 3 における内部報酬は外部報酬を本来の値よりも小さく見なすものである．具体的には，学習初期においてはゴール時の Q 値の更新はほぼ 0 の値によって更新される．つまりはどんなに高い報酬でも低い報酬でもエージェントにとってはほぼ一様のものとして認識され，それに従って Q 値が推定される．これは一見高い報酬に対して不利な変更点であるように見える．実際，この点に関しては高い報酬にとって不利に働いていることは間違いないが，外部報酬を本来よりも小さい値に見なすことには別の効果がある．それは行動の偏りの抑制である．行動の偏りはボルツマン選択によって引き起こされ，そのランダム度合いは温度定数によって制御される．温度定数は複数の Q 値をもつ行動からより高い価値を選び出す敏感さを制御しているため，Q 値の差が高いほど行動のランダム性が抑えられ，結果として行動の偏りを生む．複数報酬の問題においては，低報酬の得られるゴール（目標状態）に比べて高報酬の得られるゴールに遷移できる確率が元々低いため，高い確率で得られる報酬を基に推定した Q 値に従って行動を偏らせることで余計に高報酬

を得ることができないという悪循環が生じる．これに対して，ほぼ 0 に近いゴール報酬（内部報酬）でも，それによって複数存在する報酬全てに対して Q 値を更新する期間が設けられると，この時点での行動選択にはほとんど影響しなくとも少しずつ低下していく温度定数につられて少しずつ Q 値に従うように行動選択されるようになるため，内部報酬が小さい時点で高報酬の得られるゴールに向かうような Q 値の更新が十分にされた場合だけ高報酬を獲得できるようになると考えられる．このことから，ケース 3 の内部報酬はケース 2 よりも高いランダム性をもつ期間が存在することが高報酬獲得のための政策を学習を促したといえる．ランダム性を大きく高める方法として温度定数を限りなく高くする方法の他に，報酬をほぼ 0 にする方法があると分かった．その一方でケース 3 もケース 1，2 と同様にランダム性に依存した学習方法であることは変わらないことも明らかになった．

- ケース 4 : 提案エージェント 2 (報酬差分内部報酬エージェント)
  ケース 4 における内部報酬はそれまでのエピソードで得たゴール報酬（外部報酬）の平均値をいま得たゴール報酬から差し引いたものである．これによって，エージェントはいま得た報酬がこれまでと比べてどれだけ有用であるかという指標によって Q 値を更新する．高いゴール報酬と低いゴール報酬をどちらも得た経験があれば平均値はそれらの間を取る．そのような平均値を用いて内部報酬を計算すると，低いゴール報酬は負の値として評価されるため，Q 値も負の値に向けて更新される．Q 値の低い行動は当然選択されにくくなるため，低いゴール報酬を得ることが少なくなる．するともう一方の高い報酬のあるゴールへ到達する可能性が高まる．低い報酬のあるゴールへ行かなくなることは，同時に，その座標へ相手エージェントが進入することができるようになることを示している．この例題の場合には，自身と相手の置かれた状況がほぼ対称であることから，上のことが自身に対しても起こるため高い報酬のあるゴールへ到達できる可能性がさらに高まる．以上よりこのケースでは，元々低い報酬に陥りやすいこととに加え，高い報酬のある目標座標へ向かっても相手エージェントがいるためゴールできないことという二つの問題を解決したといえる．しかし一方では，結果でも述べたようにどちらの報酬も得られない場合があるという問題がある．この原因は Q 学習の更新式を見れば明らかである．一度低い報酬のあるゴールへ向かうように報酬が伝搬されると，ゴール座標へ到達する行動はすぐに負の値で更新されて下方に見積もられるが，ゴール座標への到達に直接関わらない行動（一つ前のステップの行動やそれ以前の行動）はすぐには下方に見積もられることはない．なぜなら，Q 値の更新には次状態における全行動に対する Q 値の内で最大をとる Q 値（$\max_{a' \in \mathcal{A}'} Q(s', a')$）だけが用いられるからである．Q 値が更新されるのは実際に選択された行動に限られるため，行動選択が Q 値の高いものから選ばれると考えると Q 値の高いものから下方に見積

もられることになる．そうして Q 値が下げられると別に Q 値の高いものが現れるといったように，上に飛び出した Q 値から順々に下げられていくことになる．この処理は高められた行動を選択し，同じように高められた次状態の最大の Q 値を用いて行う上方への更新に比べて遥かに時間がかかる．そのようにして初期座標付近まで伝搬された低い報酬のあるゴールへ向けた Q 値が一通り下げられるまでエージェントはどのゴールにもたどり着けない状況がしばらくの間続く．この状況が現れたのが実験で得られた結果であるといえる．

- ケース 5 : 提案エージェント 3 (時限増加＋平均差分内部報酬エージェント)
  ケース 5 における内部報酬はケース 4 の内部報酬をベースとして，ケース 3 の内部報酬のように最終エピソードでは通常の Q 学習と同じ更新式になるように平均報酬の影響の度合いを徐々に小さくするものである．時間による平均報酬の影響の度合いの変化を除いてケース 4 と同様であることから，特に学習初期においてはケース 4 と同じように低い報酬のあるゴールへの到達の抑制が生じる．高い報酬に対する更新は平均報酬を考慮すると元々正の値でされることに加えて，平均報酬の差し引きされる量が徐々に小さくなるため Q 値は順調に増加すると同時に，初期状態にかけて Q 値の伝搬が起こるため，どのゴールにたどり着けない状況をケース 4 よりも早く打破できたと考えられる．分かりやすく言えば，消去法をするうちに高報酬へたどり着くのを待つケース 4 に対して，報酬獲得の当てを同時に探したケース 5 の方が効率よく報酬にたどり着くことが出来たといえる．

## 5.3 実験２：学習の長さの影響

### 5.3.1 実験内容

提案手法の学習時間の長さによる影響を調べるために，$MAX\_EPISODE = 500$ と $MAX\_EPISODE = 2000$ とした実験を行った．実験ケースは実験 1 と同じものを扱う．内部報酬の中で *episode* を扱わないエージェントの結果は実験 1 の結果と合わせて時間経過とみることができる一方で，*episode* を扱うエージェント（提案エージェント 1 と 3）は *episode* を $MAX\_EPISODE$ に対する割合として扱うため，その結果は単純な時間経過とは異なることに注意されたい．実験パラメータも実験 1 と同じものを扱う．

### 5.3.2 評価指標とパラメータ設定

実験は実験 1 と同様の評価指標とパラメータ設定で実施する．各ケースでシードを変えて 100 試行を行い，最終的にエージェントが獲得し政策を評価した．具体的には，$MAX\_EPISODE$ 回目のエピソードでエージェントが獲得した報酬と，報酬獲得まで

のステップ数を評価する．表 5.3 に実験パラメータを示す．

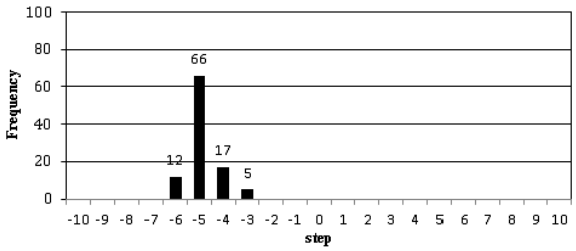
表 5.3　実験パラメータ

| | |
|---|---|
| $MAX\_STEP$ | 100 |
| $MAX\_EPISODE$ | 500 or 2000 |
| $\alpha_0$ | 0.2 |
| $\gamma_0$ | 0.9 |

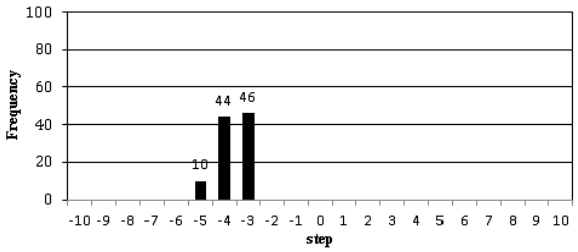
### 5.3.3　実験結果

図 5.7 と図 5.8 にそれぞれの実験結果を示す．
以下で結果について議論する．

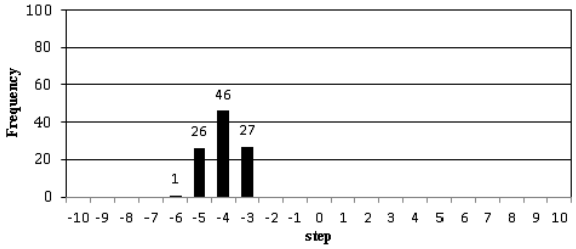
- ケース 1 : Q 学習エージェント
  図 5.7(1) と図 5.8(1) は Q 学習エージェントがエピソード数を変えても，実験 1 の結果と差異がないことを示している．これは 500 エピソード以前に学習がほとんど収束し，それ以降には他の政策を学習できなかったことを意味している．
- ケース 2 : Q 学習エージェント ($T = 1.0$)
  図 5.7(2) と図 5.8(2) は Q 学習（$T = 1.0$）エージェントが長い時間をかけて学習することでより少ないステップ数で報酬を獲得できる政策を学習できたことを示している．一方で，依然としてこのエージェントは高報酬を獲得することができないことも示された．
- ケース 3 : 提案エージェント 1 (時限増加解部報酬エージェント)
  図 5.7(3) と図 5.8(3) は $MAX\_EPISODE = 2000$ においてエージェントが全ての試行で高報酬を獲得できたことを示している一方で，$MAX\_EPISODE = 500$ においては高報酬を獲得できなかったことを示している．
- ケース 4 : 提案エージェント 2 (報酬差分内部報酬エージェント)
  図 5.7(4) と図 5.8(4) は提案エージェント 2 が高報酬を獲得する傾向をもつことを示している．特に，$MAX\_EPISODE = 2000$ においては全ての試行で高報酬を最短ステップで獲得できたことが示された．
- ケース 5 : 提案エージェント 3 (時限増加＋平均差分内部報酬エージェント)
  図 5.7(5) と図 5.8(5) は学習時間が短くてもエージェントが高報酬を獲得できたことを示している一方で，依然として最短ステップで報酬を獲得することができなかたことを示している．



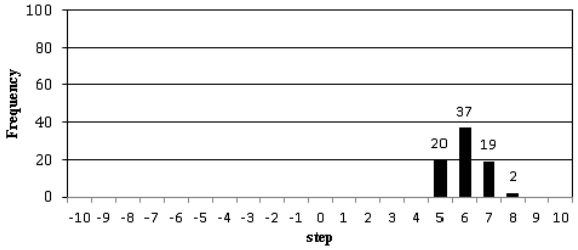

(1)Q 学習エージェント



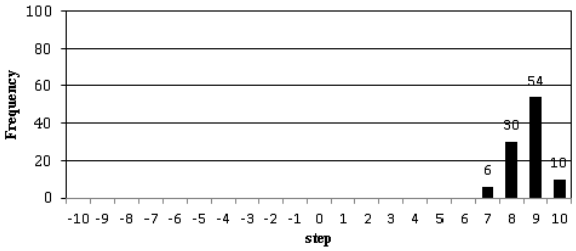

(2)Q 学習 ($T = 1.0$) エージェント




(3) ケース 3 : 提案エージェント 1 （時間増加内部報酬エージェント）

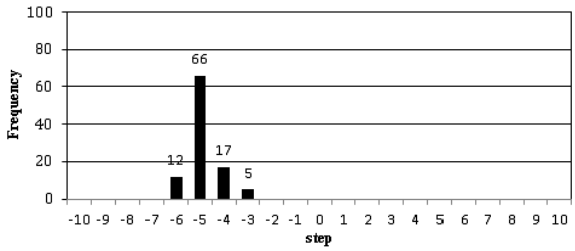



(4) ケース 4 : 提案エージェント 2 （平均差分内部報酬エージェント）

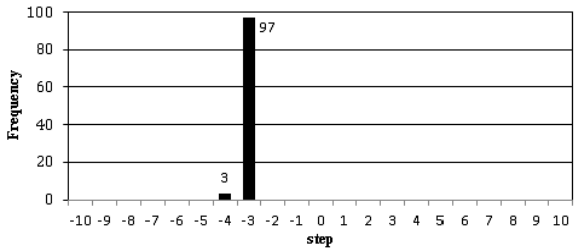



(5) ケース 5 : 提案エージェント 3 （時間増加＋平均差分内部報酬エージェント）

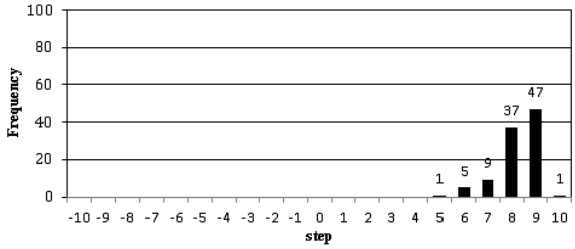
図 5.7　獲得報酬と獲得ステップ数のヒストグラム（500 エピソード学習）



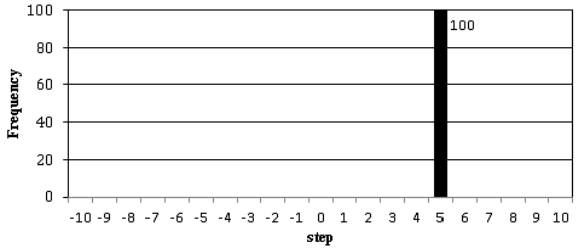

(1)Q 学習エージェント.



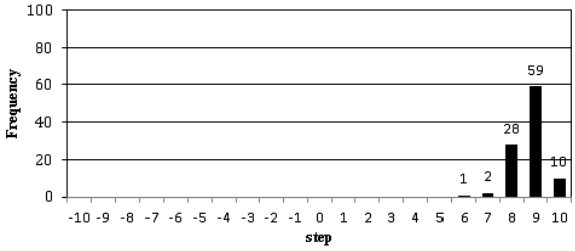

(2)Q 学習 ($T = 1.0$) エージェント




(3) ケース 3：提案エージェント１（時間増加内部報酬エージェント）




(4) ケース 4：提案エージェント２（平均差分内部報酬エージェント）




(5) ケース 5：提案エージェント３（時間増加＋平均差分内部報酬エージェント）

図 5.8　獲得報酬と獲得ステップ数のヒストグラム（2000 エピソード学習）

### 5.3.4 考察

ケース 2 の結果より，ランダム性の高い行動選択によって低報酬を獲得する周辺の経路の探索が進められたことで，より短いステップで報酬獲得可能な経路を見つけたといえる．このランダムによって稀に高報酬のあるゴールへ向かうことがあっても，ゴール座標には他のエージェントが先に到達している可能性が学習初期よりも高いため，実際に高報酬を獲得できる可能性は相当稀である．ただし，低報酬のあるゴール方向への Q 値はいつまでも上がり続けるわけではなく，温度定数 $T = 1.0$ は双方のエージェントが高報酬を獲得するための 0 ではない確率を保障している．従って，無限大の学習時間を仮定することで高報酬への政策の獲得が保障されることになる．

同じようにランダム性に頼るケースであると考察したケース 3 では, $MAX\_EPISODE$ を二倍にするだけで高報酬を得るための政策が獲得できている一方で, $MAX\_EPISODE$ が半分になったとき高報酬を得られなかったことから，高いランダム性による探索期間が十分にとれたかどうかが大きく起因していると考えられる．この例題においては, $MAX\_EPISODE = 1000$ は十分な探索期間を与えられず, $MAX\_EPISODE = 2000$ は十分な探索期間を与えることができたといえる．この内部報酬を有効にするためには適切な $MAX\_EPISODE$ をどのように設定するべきかを明らかにしなければならない．

ケース 4 の結果は実験 1 で考察したように，エージェントがどのゴールにもたどり着けない状況が時間によって解決されることが裏付けされたといえる．それよりも特筆すべきは最短ステップで高報酬を獲得するようになったことである．この理由として，何度も高報酬を獲得する内に平均報酬が限りなく高報酬に近い値になることで，内部報酬が限りなく 0 に近づくことが考えられる．0 に近い報酬での Q 値更新は他の Q 値との価値の差を小さくし，結果として温度定数による行動選択の偏りを小さくするため，周辺の探索が行われて最短経路が見つけられたと考えられる．まとめるとケース 4 の内部報酬は色々な種類の報酬を獲得する期間では低報酬を避け高報酬を探すバイアスをかけることに加え，一定の報酬ばかりを得るような段階になると平均報酬が獲得報酬に近づくことでによって Q 値は 0 に向けて更新されるようになるため探索を促す効果をもっている．

ケース 5 の結果は前の実験と大きく変わることはなかった．この理由として，学習後半に向けて通常の Q 学習と同一の学習に近づけるため，その過程でも Q 学習と同程度の（最短を探そうとしない）探索能力が発揮されたからと考えられる．他の特徴として，$MAX\_EPISODE = 500$ でも望ましい結果が得られたことより，短い期間で高報酬を確実にかぎ分けることができる能力があるといえる．上記の二点のどちらも，温度定数の適切な設定によってさらなる改善が見込まれる．

## 5.4 知見

### 5.4.1 複数報酬問題における最適政策発見の能力

複数報酬問題において最適政策を学習するために必要は能力は二つである．一つ目は低い報酬を明示的に避けて，他の可能性を陽に探索できることである．二つ目は高い報酬の周辺を探索することで最短経路を探すことができることである．一つ目は高いランダム性によっても解決可能であるが，高報酬を集中的に探索するためには平均報酬と比較して外部報酬を評価した内部報酬が有効であることが明らかになった．その内部報酬をそのまま使うよりも少しずつ増加させることでより早く高報酬を探索することが可能であることが明らかになった．二つ目は一般的なシングルエージェント強化学習と同様の課題であり，推定した Q 値に従って確定的に行動する（exploitation）とランダム性を高めて周辺を探索する（exploration）ことのバランスを取ることで効率的に解決できると考えられる．

### 5.4.2 内部報酬の適用範囲

本実験ではエージェント間の利害関係が完全に一致した例題を扱ったため，自身の問題の解決が相手も問題の解決にもつながるため，相手との情報共有などの必要性がほとんどなかった．例えば，自身の問題の解決が相手の不利益につながる問題では新たな競合が生み出される可能性がある．報酬の数が増えた場合，平均報酬が望ましくない報酬を上回ることが保証できないため，全ての低い報酬から逃れることは難しくなる．これに対しては，温度定数を操作することも考慮するなどして，必要に応じてランダム性を高くするなど，複合的な技術が必要であると予想される．その他，提案手法の一般性に関する調査は不十分である．

# 第 6 章

# パレート報酬を考慮したパレート政策探索

強化学習のフレームワークでは，通常は試行錯誤により一つの価値関数（Q テーブル）を更新することで，「一つ」の政策を獲得するが，マルチエージェント強化学習では例えば役割分担を獲得することを考えるとき多数の目標状態と報酬の組み合わせが存在するため，いくつかの報酬の組がエージェントの行動を誘引することが局所的な政策を獲得する要因となって大局的な最適政策の獲得を困難にする．また，複数のエージェントの利害関係がトレードオフを成すように完全に一致しないことが起こりうるため，最適政策が唯一つに定まらないことも考えられることから，複数あるかもしれない最適政策を一度に獲得できることも重要になる．これに対して，学習途中で見つけた有望な政策を「複数」記憶して，複数の最適政策の獲得に対応するとともに，それらを利用することにより局所的な政策に陥ることから回避して大局的な最適政策を探索するエージェントの構築を行う．多数の報酬の組み合わせのある例題として第 3 章で示したマルチステップ 4 タスク問題に適用し，提案エージェントの有効性を検証する．本章はマルチエージェント強化学習のモデル全体から見ると図 6.1 のメモリ部（黄色の部分）を対象とする．

## 6.1 提案手法 1：学習済パレート政策アーカイブの利用

### 6.1.1 概要

本章で提案するエージェントは学習過程で有望と判断した政策を保持し，それを学習中に利用することによって局所的な政策の獲得を避け，大局的な最適政策を獲得することを目指す．具体的には，(1) 確率的探索によって局所政策へ収束することを一旦は許容し，(2) 学習が十分に進められた政策をアーカイブする一方で現行の政策を初期化して学習を再開し，(3) アーカイブされた政策の情報を利用することによって同じ局所政策への収束

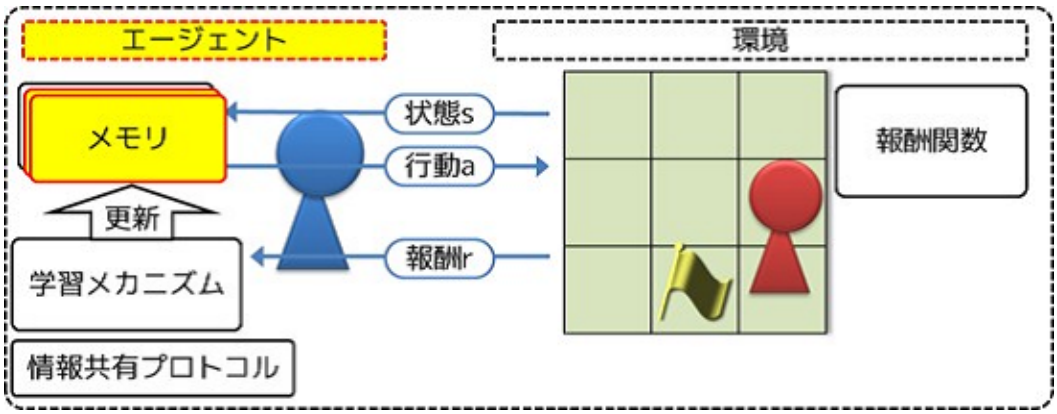



図 6.1　本章が対象とする領域

を回避しながら徐々に大局的な最適政策を獲得することを目指す．学習が進められるにつれて，アーカイブから現時点で有望な政策と考えることが出来るパレート政策以外の政策を削除していくことで，アーカイブとして保持している政策群は常にエージェントがその時点で最も有望と考える政策群であると言うことが出来る．詳しくは後述する．

### 6.1.2　パレート報酬とパレート政策

パレート最適性 [18] とは経済学で提唱された概念であり，その性質上，多目的最適化の分野やマルチエージェントシステムの領域でしばしば用いられる [6]．例えば，独立した効用をもつ二者（以上）を考えたとき，両者の効用を低下させることなく，一方の効用を高めることが出来ない状況をパレート最適であるという．以上の概念を本研究でも用いる．本論文におけるパレート政策とはパレート報酬を導く政策である．エージェントがある政策に従って行動するとき，パレート報酬を獲得できるならばその政策をパレート政策であると定義する．パレート報酬とは，エージェントが獲得した報酬の組が，各エージェントの報酬を一つの軸としたとき，全ての軸においてそれより優れている他の報酬の組がない報酬のことをいう．パレート報酬が複数あればそれらは各軸に対してトレードオフ関係を成している．例えば，報酬の組 $(16, 16)$ に対して $(20, 20)$ は各軸において優れた報酬であるため $(16, 16)$ はパレート報酬ではない．$(20, 20)$ のような関係にある報酬の組がない場合は $(16, 16)$ はパレート報酬である．そこへ他の $(17, 11), (18, 6)$ という報酬があるとき，それは片方の軸に関しては $(16, 16)$ よりも優れていないが一方の軸に関しては優れているため，これらは全てパレート報酬となる．これら複数のパレート報酬はどちらかの軸で大きい報酬をとるともう一方の軸で報酬が小さくなるトレードオフ関係を成していることがわかる．

本研究におけるパレート政策は上のようなパレート報酬を獲得した時点での政策のこと

を指す．ここで，ある政策によってある報酬がもたらされることは確率的な事象であり，その政策を利用すれば高確率で高い報酬を得ることができるわけではないことに注意されたい．重要なことはその政策によって低確率であってもその報酬が獲得できることが経験的に保障されることである．例えば，パレート報酬に対して「学習が進んでいない」状況とは，そのパレート報酬をもたらす可能性のある政策を獲得しているが，その再現性が低いことを意味する．逆に，パレート報酬に対して「学習が進んだ」状況とは，高確率でそのパレート報酬をもたらすような政策を獲得したことを意味する．

### 6.1.3 学習済みの政策

学習が進んだ状況では，高確率でそのパレート報酬をもたらすような政策を獲得したことを意味すると前項で述べた．この状況は，獲得した政策が真に良い報酬を導くものであるならば望ましい状況であると言えるが，その一方で，新しく別の報酬を探索することがほとんどなくなる．提案手法では新しく別の報酬を探索するために政策を一度初期化する方法をとるが，その直前に学習済みの政策を学習済アーカイブへ保存する．ある政策を学習済みであると判断するために次の条件を設ける．

- 全てのエージェントの政策に関するエントロピーが一定値未満のとき，その政策が学習済みであるとする．

エントロピーが低下した状態は探索の力が弱まっていることを意味する．このように確定的に行動選択を導く政策を学習済みの政策と定義する．エントロピーは第 4 章で提案したエピソードに基づくエントロピーを用いる．

### 6.1.4 アーキテクチャ

提案エージェントのアーキテクチャについて述べる．エージェントは複数の政策を保持するため，本来一つだけ持つ $Q(s,a)$ のテーブルを必要な分だけ持つ．必要な数は問題環境における報酬の組み合わせの数に依存する．最悪の場合で報酬の組み合わせの数と同じだけ $Q(s,a)$ のテーブルを保持する．各 $Q(s,a)$ のテーブルには，保存したエピソード時間とそのエピソードで獲得した報酬の値を付随して保持する．さらに，他エージェントとの通信機能，仮想エピソード数という変数を新しく備える．

### 6.1.5 メカニズム

適用の流れは図 6.2 に示す通りである．基本的な方針は，学習完了したと判断したら政策を学習済アーカイブへ保存し，政策を初期化して学習を続行することを繰り返す．エピ

ソードの終わりに，そのエピソードで得た報酬が学習済アーカイブの報酬より全ての軸に対して劣っていないかを確認し，劣っている場合はそのエピソードの学習を無効にする．こうして，学習済みの政策を凌ぐ可能性のある政策に対する学習のみ有効にし，獲得を促進する．

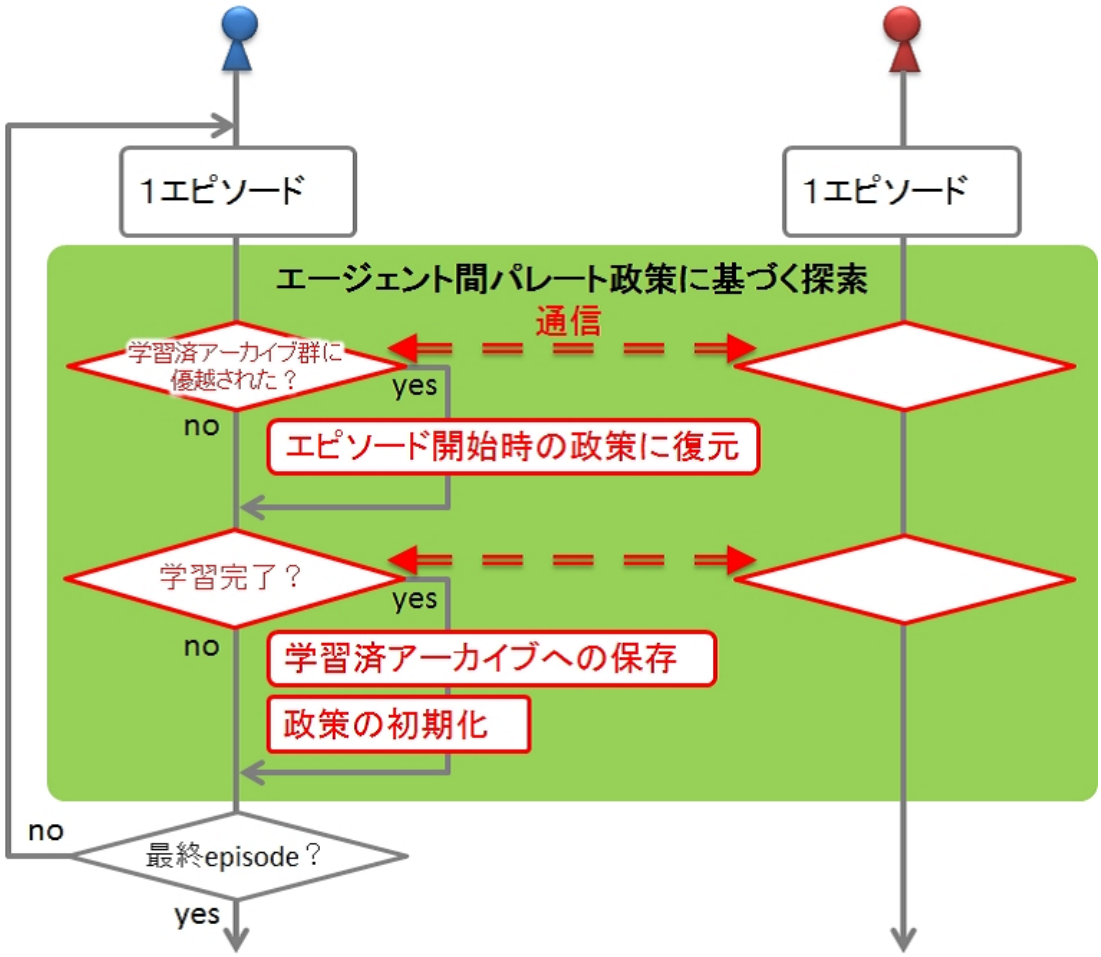



図 6.2　学習済パレート政策アーカイブの利用エージェントの処理の流れ

1. 学習済アーカイブとの比較
   該当のエピソードで得た報酬値と学習済アーカイブ内の政策に付随する報酬値を比較し，このエピソードにおいて獲得した報酬が学習済アーカイブ全ての報酬に優越されていないか（パレートであるかどうか），通信を用いて他のエージェントと情報を共有して調べる．もし，学習済アーカイブのいずれかの報酬に優越されている（パレートでない）場合，このエピソードにおける学習が非パレート政策に対する学習であったと判断し，学習を無効にするため，$Q(\mathcal{S}, \mathcal{A})$ をエピソードの初めの値に戻す．エピソード数に応じて変化するボルツマン選択の温度パラメータ $T$ を学

習が無効になった期間と同様に変化させないため，温度パラメータ算出のために用いる仮想エピソード数（$episode_{virtual}$）も増加させないようにする．

2. 学習完了政策の学習済アーカイブへの保存
   パレート報酬を獲得した経験（エピソード）のみによって学習が進められる政策は，新しい学習済みのパレート政策となる．エピソードの終了時に現政策が学習完了の条件を満たした場合，現在の $Q(\mathcal{S},\mathcal{A})$ をアーカイブとして保存する．ここで報酬値と現在のエピソード数も併せて保存し，後のエピソードでのパレートの判断に用いる．これに加えて，温度パラメータ算出のために用いる仮想エピソード数（$episode_{virtual}$）も併せて保存し，後に政策を再現する際に用いる．新しい政策がそれまでアーカイブに保存されていた政策を優越した場合は，パレート政策でなくなった $Q(\mathcal{S},\mathcal{A})$ は報酬値，仮想エピソード数と併せてアーカイブから削除する．

アーカイブ保存の例を図 6.3 に示す．(a) ではまだいずれの政策も保存されていない状態であったため，無条件でアーカイブ保存する．(b) では $(4,5)$ という報酬を得られた政策がどちらのエージェントの観点からも $(2,3)$ よりも優れているためアーカイブ保存する．この影響で $(2,3)$ はパレートではなくなったため，報酬 $(2,3)$ をもたらした政策はアーカイブから削除する．(c) は $(7,3)$ という報酬を得たエピソードでこの報酬の組はエージェント A の観点から優れた政策であるため，この政策はアーカイブ保存される．(d) は (c) までに追加されたアーカイブが学習を抑制する領域を示している．このため，次に学習が完了する政策はこの領域の外側，つまり新たなパレート政策となる．

以上のように，優れた報酬を導く可能性のある政策のみを保存し，そのような政策を得たエピソードの $Q(s,a)$ の更新のみを認めることで局所政策へ陥ることを慎重に避けることができるため，結果として大局的な最適政策を獲得されやすくなることが期待される．

最後に学習済パレート政策アーカイブに基づくエージェントのアルゴリズムを **Algorithm 3(1)** に示す．全大文字は定数を表しており，$Env$ は環境の略称でありドットに続く情報はエージェントではなく環境が保持している．ID はエージェントの識別符号，$\vec{r}$ は報酬の組，$IsPareto$ は共有した報酬の組がアーカイブの中でパレートになるかどうか調べる関数，$Nop$ は何も処理しないことを表す関数，$IsLearned$ はその政策が学習完了したかどうか判定する関数，$\theta_{learned}$ は学習完了条件の閾値，$DeleteNonparetoPolicyFromLearnedArchives()$ は学習済アーカイブから非パレートになった政策を削除する関数を表している．

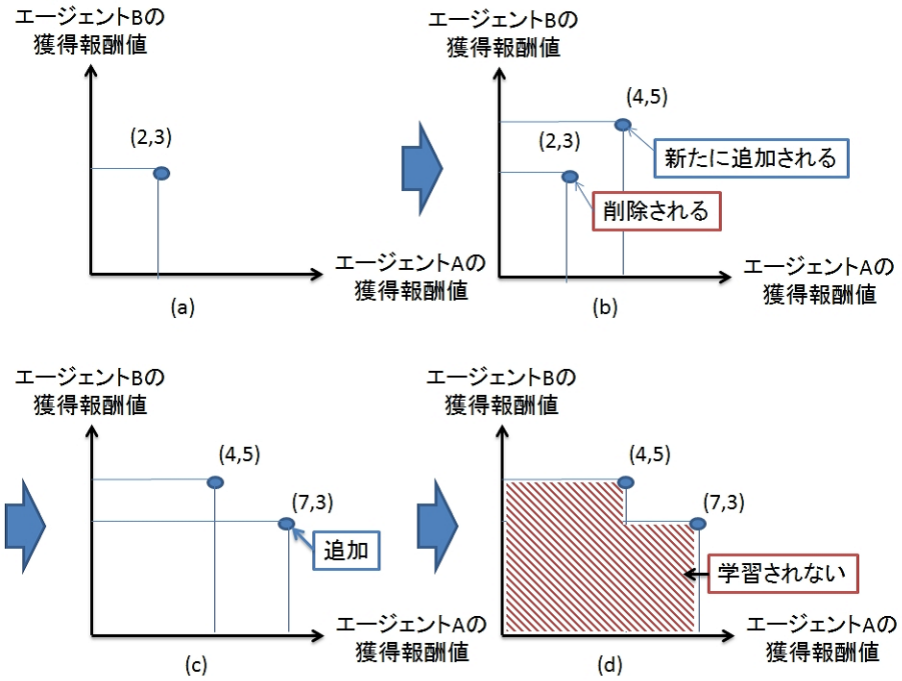



図 6.3　エージェント間パレート政策のアーカイブ保存の例

**Algorithm 3(1)** Q-learning based on learned pareto policy archives

$\alpha \leftarrow \alpha_0 \quad \gamma \leftarrow \gamma_0$
$episode \leftarrow 0$
$episode_{virtual} \leftarrow 0$
**while** $episode <$ `MAX_EPISODE` **do**
  $s \leftarrow Env.StartState(\mathrm{ID})$
  $step \leftarrow 0$
  $Q_{prev}(\mathcal{S}, \mathcal{A}) \leftarrow Q(\mathcal{S}, \mathcal{A})$
  **while** $step <$ `MAX_STEP` $ors \not\equiv Env.GoalState(\mathrm{ID})$ **do**
    $StoreState(s)$
    $a \leftarrow SelectAction(Q(\mathcal{S}, \mathcal{A}), episode_{virtual})$
    $DoAction(a)$
    $s' \leftarrow Env.State(\mathrm{ID})$
    $r \leftarrow Env.Reward(s')$
    $Q(s, a) \leftarrow UpdateQ(r, s', A', \alpha, \gamma)$
    $step \leftarrow step + 1$
    $s \leftarrow s'$
  **end while**
  $r(\texttt{OTHERS\_ID}) \leftarrow Communicate(\texttt{OTHERS\_ID}, r)$
  **if** $IsPareto(\vec{r})$ **then**
    $Nop$
  **else**
    $Q(\mathcal{S}, \mathcal{A}) \leftarrow Q_{prev}(\mathcal{S}, \mathcal{A})$
    $r_{virtual} \leftarrow episode_{virtual} - 1$
  **end if**
  $h \leftarrow CalculateEntropy()$
  $H(\texttt{OTHERS\_ID}) \leftarrow Communicate(\texttt{OTHERS\_ID}, h)$
  **if** $IsLearned(Q(\mathcal{S}, \mathcal{A}, episode_{virtual}, \theta_{learned})$ **then**
    $StorePolicyAsLearnedArchives(Q(\mathcal{S}, \mathcal{A}, episode_{virtual}, r)$
    $DeleteNonparetoPolicyFromLearnedArchives()$
  **end if**
  $episode \leftarrow episode + 1$
  $episode_{virtual} \leftarrow episode_{virtual} + 1$
**end while**

## 6.2 提案手法２：学習中パレート政策アーカイブの利用

### 6.2.1 概要

前節で示した提案手法１は学習完了したパレート政策のみを扱ったが，学習完了以前のパレート政策を提案手法１と同様に利用することで，非パレート政策への学習を強く抑制することが可能であると考えられる．これにより，最終的にパレート政策でなくなる無駄な政策の学習が省略され，より早く効率的な学習が期待できる．

### 6.2.2 学習済パレート政策と学習中パレート政策の扱い

本手法では学習済パレート政策は学習済アーカイブとして，学習中パレート政策は学習中アーカイブとして保持する．どちらのアーカイブも優越する報酬を得た学習を無効にすることで，そのような局所政策の獲得を抑制する．ただし，次の点において学習済パレート政策と学習中パレート政策の扱いは異なる．学習完了した政策は，これと同じ政策（同じ報酬（値）を得る政策）を再び学習することを避けるため，全く同じ報酬値であった場合の学習も無効の対象とする．一方で，学習完了していない政策は，この政策に対する学習を引き続き進めるために，全く同じ報酬値であった場合の学習は無効にしない．

### 6.2.3 アーキテクチャ

提案手法２のエージェントアーキテクチャは提案手法１とほぼ同一である．エージェントは複数の政策を保持するため，本来一つだけ持つ $Q(s,a)$ のテーブルを必要な分だけ持つが，それらは「学習済アーカイブ」と「学習中アーカイブ」に明確に分けられて保存される．

### 6.2.4 メカニズム

適用の流れは図 6.4 に示す通りである．基本的な方針は，学習完了したと判断したら政策を学習済アーカイブへ保存し，政策を初期化して学習を続行することを繰り返す．エピソードの終わりに，そのエピソードで得た報酬が学習済アーカイブの報酬より全ての軸に対して劣っていないかを確認し，劣ってい場合はそのエピソードの学習を無効にする．これに加えて，学習完了の条件を満たしていない政策でもパレート政策であると判断した政策を学習中アーカイブへ保存する．こうして，学習済み，学習中のパレート政策を凌ぐ可能性のある政策に対する学習のみ有効にし，獲得を促進する．

1. 学習済アーカイブとの比較

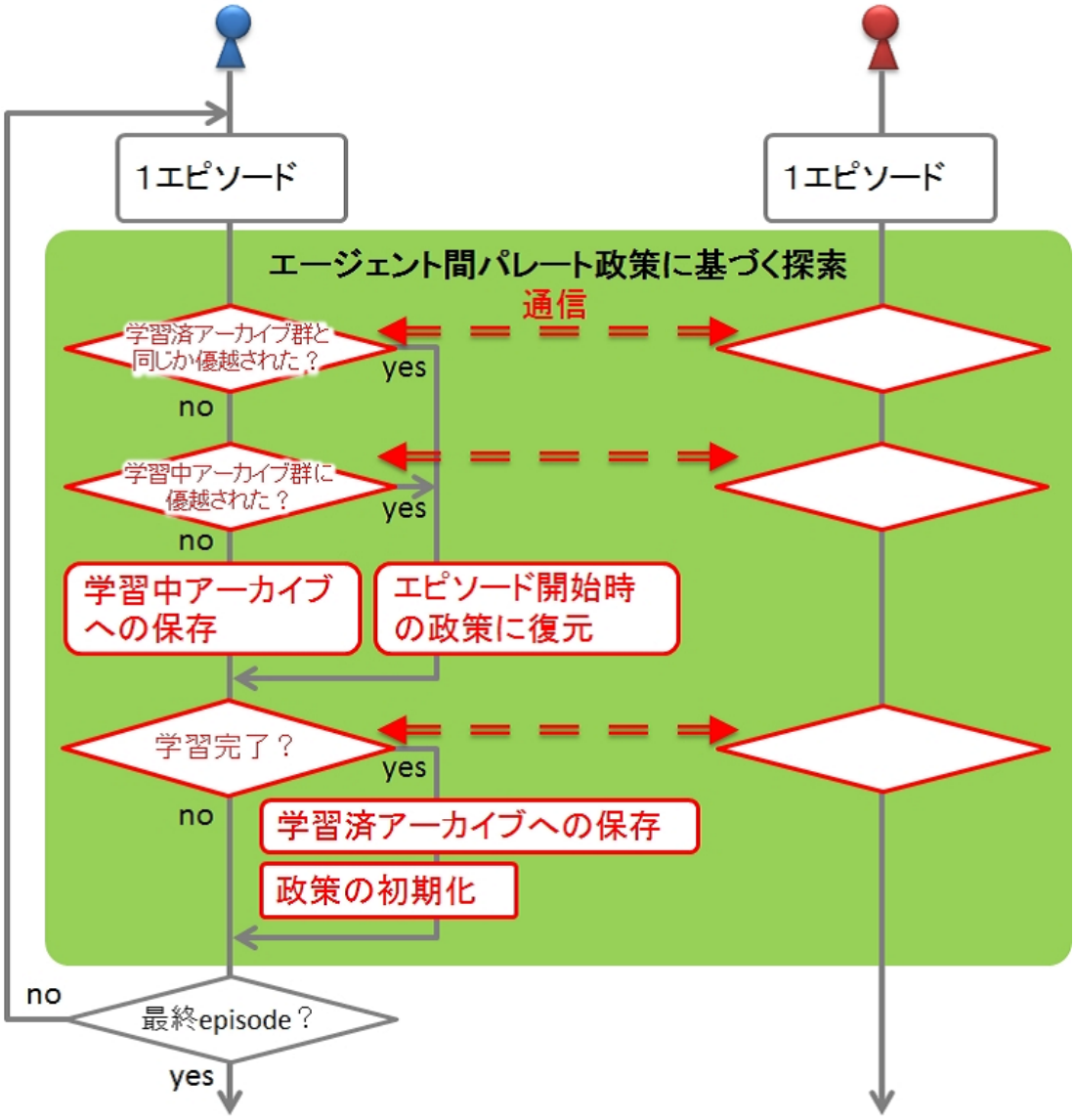



図 6.4　学習済＋学習中パレート政策アーカイブの利用エージェントの処理の流れ

該当のエピソードで得た報酬値と学習済アーカイブ内の政策に付随する報酬値を比較し，このエピソードにおいて獲得した報酬が学習済アーカイブ全ての報酬に優越されていないか（パレートであるかどうか），通信を用いて他のエージェントと情報を共有して調べる．もし，学習済アーカイブのいずれかの報酬に優越されている（パレートでない）場合，このエピソードにおける学習が非パレート政策に対する学習であったと判断し，学習を無効にするため，$Q(\mathcal{S}, \mathcal{A})$ をエピソードの初めの値に戻す．エピソード数に応じて変化するボルツマン選択の温度パラメータ $T$ を学習が無効になった期間と同様に変化させないため，温度パラメータ算出のために用いる仮想エピソード数（$episode_{virtual}$）も増加させないようにする．

2. 学習中アーカイブとの比較
   学習中アーカイブに対しても前項と同じ処理を行う．ただし，学習中パレート政策の扱いについて述べた通り，この政策に対する学習は引き続き進めるために，全く同じ報酬値であった場合の学習は無効にしない．
3. 学習中アーカイブへの保存
   上二項のどちらにおいても学習が無効にされなかった場合，有用な学習がなされたと判断できるため，現在の政策を学習中アーカイブとして保存する．ここでの処理は次項の学習済アーカイブへの保存と同様である．
4. 学習完了政策の学習済アーカイブへの保存
   パレート報酬を獲得した経験（エピソード）のみによって学習が進められる政策は，新しい学習済みのパレート政策となる．エピソードの終了時に現政策が学習完了の条件を満たした場合，現在の $Q(\mathcal{S},\mathcal{A})$ をアーカイブとして保存する．ここで報酬値と現在のエピソード数も併せて保存し，後のエピソードでのパレートの判断に用いる．これに加えて，温度パラメータ算出のために用いる仮想エピソード数（$episode_{virtual}$）も併せて保存し，後に政策を再現する際に用いる．新しい政策がそれまでアーカイブに保存されていた政策を優越した場合は，パレート政策でなくなった $Q(\mathcal{S},\mathcal{A})$ は報酬値，仮想エピソード数と併せてアーカイブから削除する．

学習中アーカイブを含めたアーカイブの利用イメージを図 6.5 に示す．(a) では $(4,5)$ という報酬を得る学習済アーカイブの政策が学習を無効にする領域を示している．(b) では $(7,3)$ という報酬はエージェント A の観点から優れた政策であるため，この政策は学習中アーカイブに保存される．(c) は獲得報酬が $(7,3)$ となる学習中アーカイブの政策が新たに学習を無効にするようになる領域を示している．提案手法１では $(7,3)$ という報酬を得る政策が学習完了するまで上記の領域が学習無効になることはないため，手法２ではその過程が省略され，無駄な学習が削減されると言える．

最後に学習済・学習中パレート政策アーカイブに基づくエージェントのアルゴリズムを **Algorithm 3(2)** に示す．全大文字は定数を表しており，$Env$ は環境の略称でありドットに続く情報はエージェントではなく環境が保持している．ID はエージェントの識別符号，$\vec{r}$ は報酬の組，$IsPareto$ は共有した報酬の組がアーカイブの中でパレートになるかどうか調べる関数，$Nop$ は何も処理しない関数，$IsLearned$ はその政策が学習完了したかどうか判定する関数，$\theta_{learned}$ は学習完了条件の閾値，$DeleteNonparetoPolicyFromLearnedArchives()$ は学習済アーカイブから非パレートになった政策を削除する関数を表している．なお，$LearnedArchives$ と $LearningArchives$ はそれぞれ学習済アーカイブと学習中アーカイブを指す．

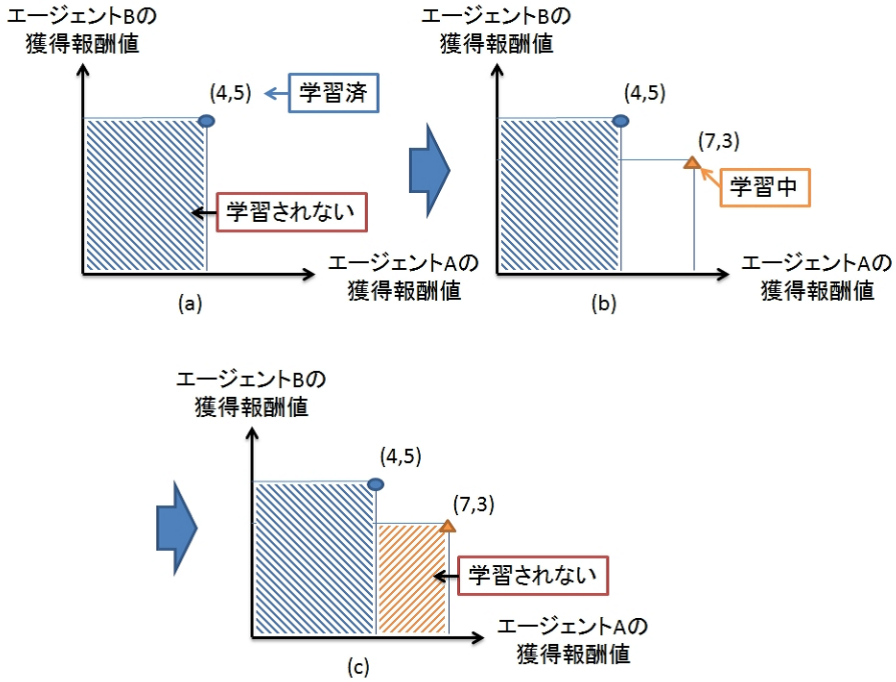



図 6.5　学習済・学習中アーカイブの利用イメージ

**Algorithm 3(2)**

Q-learning based on learned / learning pareto policy archives

$\alpha \leftarrow \alpha_0 \quad \gamma \leftarrow \gamma_0$

$episode \leftarrow 0$

$episode_{virtual} \leftarrow 0$

**while** $episode <$ `MAX_EPISODE` **do**

  $s \leftarrow Env.StartState(\mathrm{ID})$

  $step \leftarrow 0$

  **while** $step <$ `MAX_STEP` $ors \not\equiv Env.GoalState(\mathrm{ID})$ **do**

    $StoreState(s)$

    $a \leftarrow SelectAction(Q(\mathcal{S}, \mathcal{A}), episode_{virtual})$

    $DoAction(a)$

    $s' \leftarrow Env.State(\mathrm{ID})$

    $r \leftarrow Env.Reward(s')$

    $Q(s, a) \leftarrow UpdateQ(r, s', A', \alpha, \gamma)$

    $step \leftarrow step + 1$

    $s \leftarrow s'$

  **end while**

  $r(\texttt{OTHERS\_ID}) \leftarrow Communicate(\texttt{OTHERS\_ID}, r)$

  **if** $IsParetoForLearnedArchives(\vec{r})$

    **and** $IsParetoForLearningArchives(\vec{r})$ **then**

    $StorePolicyAsLearningArchives(Q(\mathcal{S}, \mathcal{A}, episode_{virtual}, r)$

    $DeleteNonparetoPolicyFromLearningArchives()$

  **else**

    $Q(\mathcal{S}, \mathcal{A}) \leftarrow Q_0(\mathcal{S}, \mathcal{A})$

    $r_{virtual} \leftarrow episode_{virtual} - 1$

  **end if**

  $h \leftarrow CalculateEntropy()$

  $H(\texttt{OTHERS\_ID}) \leftarrow Communicate(\texttt{OTHERS\_ID}, h)$

  **if** $IsLearned(Q(\mathcal{S}, \mathcal{A}, episode_{virtual}, \theta_{learned})$ **then**

    $StorePolicyAsLearnedArchives(Q(\mathcal{S}, \mathcal{A}, episode_{virtual}, r)$

    $DeleteNonparetoPolicyFromLearnedArchives()$

  **end if**

  $episode \leftarrow episode + 1$

  $episode_{virtual} \leftarrow episode_{virtual} + 1$

**end while**

## 6.3 実験

### 6.3.1 実験内容

提案エージェントの有効性を検証するため，3.3 章で示したマルチステップ 4 タスク問題を計算機上に実装し，表 6.1 に示すケースに分けて実験した．ケース 1 は通常の Q 学習エージェント，ケース 2，ケース 3 をそれぞれ提案手法 1，提案手法 2 のエージェントとした．

表 6.1　実験ケース

| ケース | エージェント |
|---|---|
| ケース 1 | Q 学習エージェント |
| ケース 2 | 学習済パレート政策に基づくエージェント |
| ケース 3 | 学習済・学習中パレート政策に基づくエージェント |

### 6.3.2 評価指標とパラメータ設定

実験は各ケースでシードを変えて 100 試行を行い，最終的にエージェントが獲得した政策を評価した．具体的には，$MAX\_EPISODE$ 回の繰り返しの間にエージェントが獲得した政策の内容を評価する．表 6.2 に実験パラメータを示す．$MAX\_STEP$ はエージェントが一向に目標状態に辿り着かない場合に学習を打ち切るステップ数である．

表 6.2　実験パラメータ

| | |
|---|---|
| $\theta_{learned}$ | 0.5 |
| $MAX\_STEP$ | 100 |
| $MAX\_EPISODE$ | 50000 |
| $\alpha_0$ | 0.1 |
| $\gamma_0$ | 0.95 |

### 6.3.3 実験結果

ケース 1 の Q 学習エージェントは 100 試行全てで報酬 $(4, 4)$ へ向かう政策を獲得する結果となった．一方でケース 2，ケース 3 では，100 試行全てでパレート報酬へ向か

う政策全てを獲得できた．具体的には，二体のエージェントが得る報酬の組が $(18,6)$, $(6,18)$，$(17,11)$，$(11,17)$，$(16,16)$ となる政策が最終エピソードまでに学習済アーカイブに保存されていた．報酬 $(4,4)$ は全てのエージェントがスタートから最も近いゴールである．報酬 $(16,16)$ は全てのエージェントの報酬最大化しようとしたとき導かれるものである．報酬 $(18,6)$，$(6,18)$，$(17,11)$，$(11,17)$，$(16,16)$ は二体のエージェントの得る報酬という観点からそれ以上優れたものがない真のパレート報酬である．ケース 2 とケース 3 で上記のパレート政策全て獲得するのにかかったエピソード数に関して表 6.3 と図 6.6 に示す．表は 100 試行の全パレート政策が学習済みとなるまでにかかったエピソード数に関するデータである．図は 1000 エピソードの区間で見たとき，100 試行中各区間に該当するエピソード数で学習済みとなった回数を表している．青色がケース 2，赤色がケース 3 の結果を示すグラフである．図を見てよくわかる通り，頻度が多いピークと，学習完了にかかるエピソード数の広がりに差異がみられる．結果的に学習完了にかかるエピソード数の平均はほとんど違いがなかった．

表 6.3　全パレート政策獲得にかかったエピソード数 (100 試行)

| | ケース 2 | ケース 3 |
|---|---|---|
| 平均エピソード数 | 21808.81 | 21098.27 |
| 最小エピソード数 | 17366 | 13785 |
| 最大エピソード数 | 25882 | 43737 |
| エピソード数の標準偏差 | 1785.34 | 4269.48 |

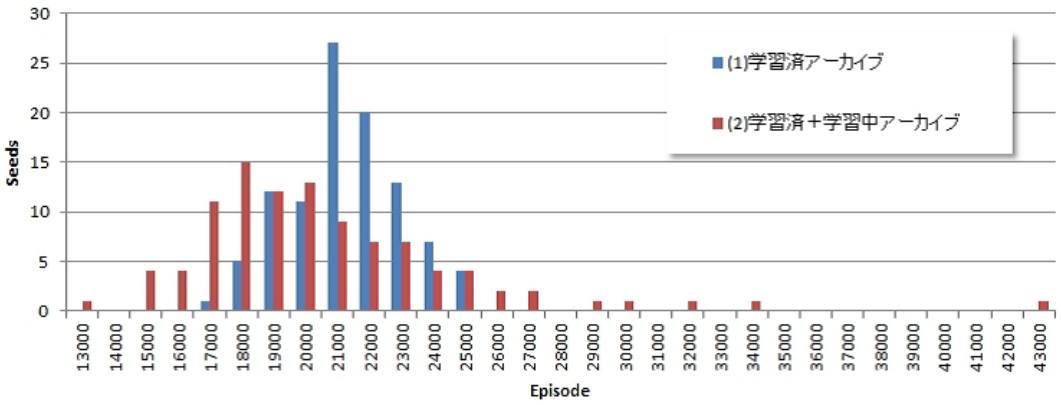



図 6.6　全パレート政策獲得にかかったエピソード数のヒストグラム (100 試行)

### 6.3.4 考察

**探索行動の強化**

強化学習では，一つあるいはたくさんのエピソードの経験によって得た知識（ここでは $Q(\mathcal{S},\mathcal{A})$）を利用しながら効率的に報酬が得られる政策を強めること（exploitation と呼ばれる）と，新しいより良い報酬（あるいは単位時間当たりの獲得報酬の最大化）を探すこと（exploration と呼ばれる）をバランスよく行うことが重要であると議論されてきた．しかし実際は，一つの有用な報酬だけに注目したほうが効率が上がるかも知れないし，未知の環境での学習が前提となる状況で良いバランスを取ることは困難である．ケース 1 の結果が全く望ましいものにならなかった原因の一つとして上記のバランスが取れていないことが考えられる．Exploitation が強すぎたため，初期に得られた報酬の方へ向かう政策が強められたことが原因である．では逆に，最大限に exploration を強めた場合，つまり完全にランダムな行動選択によって良い政策が得られるかといえば，実際に実験したところ少なくとも 50000 エピソードでは報酬 $(16,16)$ を得るような良い政策を獲得することができず，せいぜい報酬 $(12,12)$ へ向かう政策を獲得するという結果が得られた．これによってわかることは，極端に到達確率の低い目標状態へ向かうためには，そこへ到達したエピソードの経験によって得た知識を利用することが重要であることである．これに対して，通常の Q 学習は全ての報酬に対して一様にしか知識を構成できず，選り好みして学習するような機構を持たないため，単純に知識利用を強めようとする調整は，より到達確率の高い目標状態に対しても一様に働いてしまうため，到達確率にいっそうの差が生まれてケース 1 のような結果が生じることになる．提案手法 1，2 では上記の知識利用の差別化を明示的に行っているといえる．学習が十分に進んだ政策によって得られる報酬（とそれに劣る報酬）に対しては，それ以上の学習が必要ないものとして，これに対する知識を蓄えないことで，必要な知識だけ蓄積，利用することが可能となった．

**パレート政策獲得の速さ**

全てのパレート政策が学習済みとなるまでにかかったエピソード数に関してはケース 1 とケース 2 の間で興味深い差異が見られた．ケース 2 に関してはほぼ偏りのない正規分布になっているように見える．学習済アーカイブのみ利用するケース 2 では到達確率の高い目標状態から順（これは表 6.7 を参照）に，到達確率と学習完了までに必要な到達回数に従った結果と言える．実際は経験に基づく知識の利用によって到達確率は徐々に増加する．ケース 3 において学習完了にかかったエピソード数のピークがケース 3 よりも低い位置に変わった理由は学習中アーカイブの効果によるものであろう．これは学習中アーカイブが期待通りにいくつかの報酬に対する無駄になる学習を抑制したからである．一方で，学習完了に膨大な時間を費やすことがあることが見られた．これは確率的な外れ値である

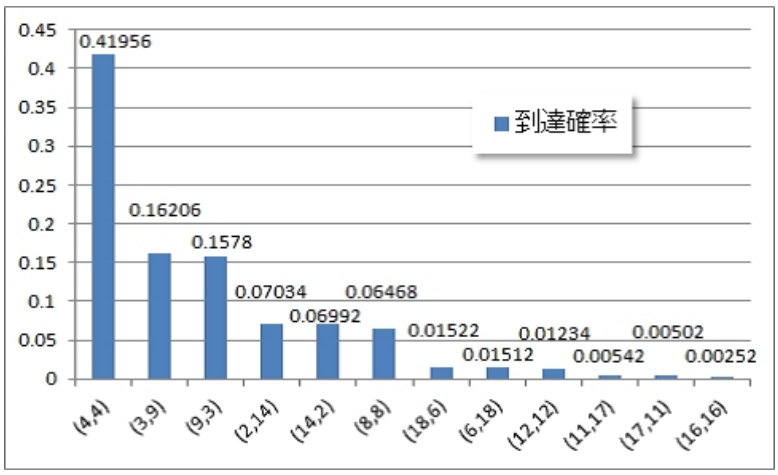



図 6.7　例題におけるランダムな行動選択下での各目標状態への到達確率

というより，分布に正確に従っているように見える．この原因として考えられるのが，ほとんど行動選択が偏った段階（かつ学習完了に判定されないくらい）で初めてそれを優越するパレート政策が現れることが挙げられる．このれによってほぼ確定し始めていた政策に従って行動選択される一方で，非パレートとなったその報酬に対してそれ以上学習が許されなくなると，別のパレート報酬を得られる報酬に対して低い到達確率によって少なくなった学習機会に学習するしかなくなることが問題となる．この結果にから，学習が完了していないパレート政策のアーカイブを利用して，学習を許さない領域を作ることの危険性が明らかになった．

**パレート政策の学習の意味**

例題における報酬の組のプロットを図 6.8 に示す．$(6,18)$，$(18,6)$ や $(17,11)$，$(11,17)$ はパレート報酬ではあるが，全てのエージェントが得た報酬の報酬の総和という観点からは最適とはいえない．しかし，報酬の大小はそれぞれのエージェント内部では比べられるものだとしても，エージェント間では優劣をつけられないという考え方もある．どちらの考え方が正しいということは一定に決められることではない．例えば，このエージェントの設計者が，全てのエージェントが獲得する報酬の総和を最大化することを目的として設計したのであれば $(16,16)$ が最適であることは間違いない．一方でエージェントごとに異なる稼働コストがあるような問題をモデル化したのであれば，パレート政策を求めることは有用である．これに対して稼働コストを定量化し報酬関数に組み込むことも可能であるが，報酬関数が複雑なモデルになれば大局的な最適政策の獲得が難しくなることにも注意しなければならない．報酬の総和を最大化が基準となる場合はともかく，単純な計算式に落とし込めない評価が基準となる問題では，当然エージェントにその判断を任せることができない．単一の政策の学習を前提とする通常の Q 学習では，評価をエージェントに任

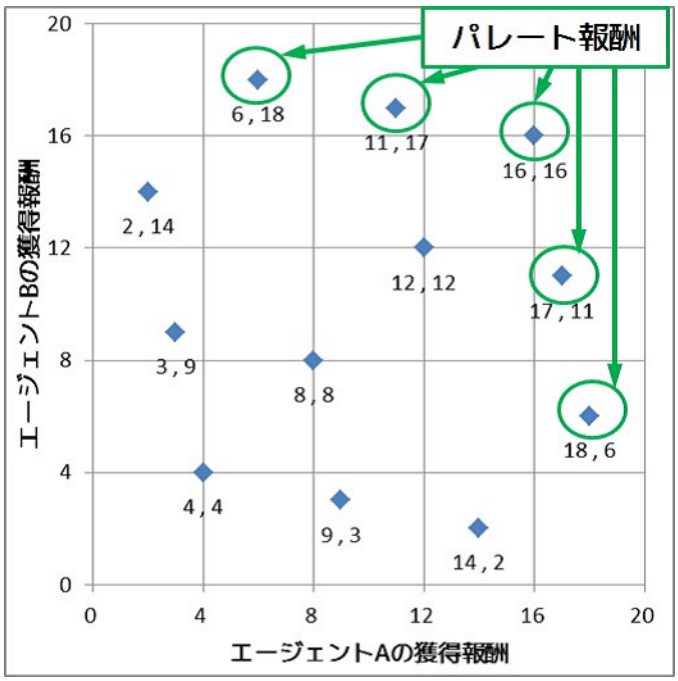



図 6.8　例題における全報酬の組のプロットとパレート報酬

せるために基準を実装しなければならないが，設計者が正しい基準を実装することは困難である．この場合，最終的には設計者である人間が判断することが必要になるが，そのためにはエージェントが候補となる政策群を学習出来ることが重要であると言えます．選択肢の中から選ぶ方式は，予め選択基準を決めなければならない場合より，人間の負担を大きく軽減できる点で非常に有用であるといえる．

**複数政策の保持の応用**

提案手法は様々なバリエーションのある政策の中から真に必要なものに対して排他的に知識利用をすることを可能にする方法であると上述した．本論文においては真に有用な政策を「パレート報酬を得るための政策」としたことで，該当する政策を漏れなく学習できたに過ぎない．同様の観点から考えると真に有用な政策（設計者にとって望ましい政策）を様々なものに設定することが可能で，それによって該当する政策を漏れなく学習できるはずである．さらに複数の政策を保持する本手法においては必ずしも一意に定まるように厳密に設定する必要がないことは，繰り返しになるが，人間の負担を大きく軽減できる点で大きな利点となる．

# 第 7 章

# 提案手法の統合に関する調査

## 7.1　モデルに内在する問題への具体的な対処とその課題

本研究は，マルチエージェント強化学習において，学習中のエージェント間の複雑な相互作用が原因でモデル化の段階では自覚することが困難な問題をモデルに内在する問題と称して着目し，エージェントのモデルを改善することによってモデルに内在する問題（特に同時学習問題や報酬の組み合わせが増えることで表面化する問題）の解決を図った．それぞれの問題に対応する例題への取り組みの中で，次の知見を得た：(1) 一定の学習速度のモデルに内在する問題に対しては，通信によって共有した学習進度を基にして学習パラメータ（特に，割引率 $\gamma$）を調整することでエージェント間で学習の阻害を回避できることを明らかにした．(2) 報酬の受容のモデルに内在する問題に対しては，複数報酬の環境において与えられた報酬値をそのまま受け取り学習せず，外部報酬から見積もった内部報酬（特に今までの獲得報酬の平均値を差し引いた値）を用いて Q 値を更新することで高い報酬を探索できることを明らかにした．(3) 単一政策の保持のモデルに内在する問題に対しては，学習途中に見つけた有望なパレート政策をアーカイブ保存し，それに基づいて状態-行動価値の更新を決定することによって，局所政策の獲得を回避できることを示した．これらの問題解決のモチベーションは，与えられる報酬に対して適切に学習するために，その反対の状況を打破するというアプローチにより解決が試みられている点で共通している．もしもこれらの問題が全て起こるモデルに対するならば，図 7.1 に示すように下位の問題から順に全てを解決していくことが必要である．まず，報酬が得られないという最悪な状況を打破するために，学習進度に基づく競合回避手法によってとにかく報酬を探索することが必要である．次に，できるだけより良い政策の獲得のため高い報酬を探索することが必要である．さらに，望ましい報酬の探索を阻害するような報酬の組をエージェント間のパレート報酬という観点から区別して，局所政策に陥らないように，かつ効率的に探索することが必要である．しかし，モデルに内在する問題は性質上，いつ，どのような問題が引き起こされるか予測ができない．例えば，図 7.2 に示すように全て同時に起こ


図 7.1　モデルに内在する問題への段階的な対処のイメージ

るモデルばかりではなく，モデルによって二つが同時に起こる場合，あるいは一つも起こらない場合も考えられる．そのため，考えられ得る様々な問題全てについて対処法を検討し，それら全てを盛り込んだ統合的な対処法を探究することが重要である．

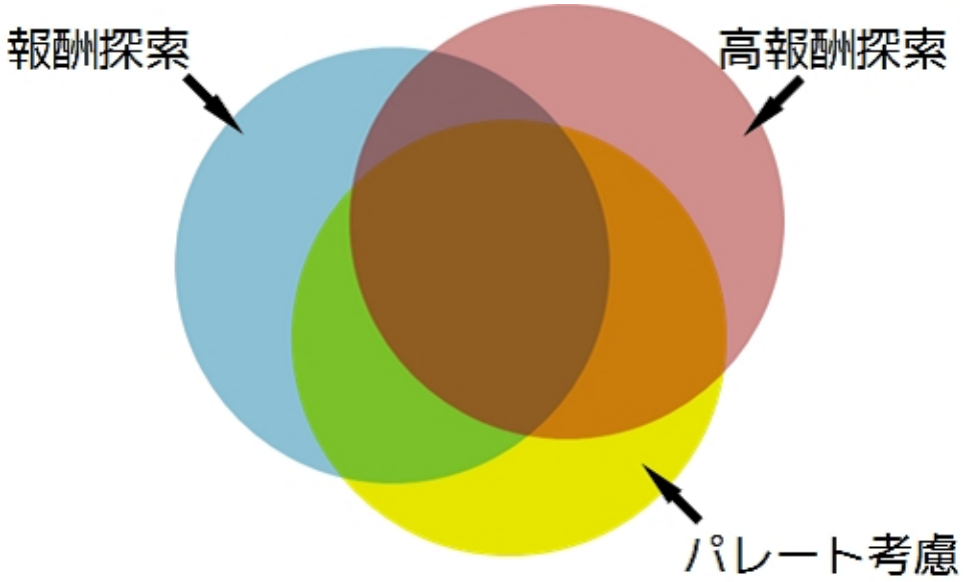



図 7.2　モデルに内在する問題とその対処の重ね合わせのイメージ

そこで本章では，統合に関する実例調査として，前章までに提案した三つの手法を統合して行った検証について述べる．統合の対象としたのは，各提案手法毎で特に有効性が認められたものに限り，学習進度に基づく学習速度の調整による競合回避手法からは学習が進んでいる（あるいは遅れている）エージェントの割引率を減少（あるいは増加）させる手法を，内部報酬に基づく大域的最適解探索手法からは平均差分内部報酬と時限増加平均

差分内部報酬を，パレート報酬を考慮したパレート政策探索手法からは学習済アーカイブだけ利用した手法と学習中アーカイブも利用した手法である．

## 7.2 実験

### 7.2.1 実験内容

独立した三種のメカニズムを同時に適用することでエージェントにどのような影響をもたらすかを調査するため，三種の提案手法で特に有効性が認められたものを組み合わせて，第 3.3 章で示し，第 6 章の実験でも用いたマルチステップ 4 タスク問題を同様に計算機上に実装し，いくつかのケースに分けて実験した．以後便宜的に三種の提案手法をそれぞれカテゴリ 1，2，3 と呼ぶことにする．実験ケースはカテゴリ 1〜3 のそれぞれから方法を一つずつ選び出し，三つを組み合わせたものを扱う．具体的には下記に示す各カテゴリ毎に三つずつ，それらを組み合わせた計 27 ケースとした．

- カテゴリ 1 :
  1. 適用なし
  2. 学習の進んでいるエージェントの割引率を下げる（$\theta = 0.6, d\gamma = 0.6$）
  3. 学習の遅れているエージェントの割引率を上げる（$\theta = 0.0, d\gamma = 1.05$）
- カテゴリ 2 :
  1. 適用なし
  2. 平均差分内部報酬
  3. 時限増加平均差分内部報酬
- カテゴリ 3 :
  1. 適用なし
  2. 学習済アーカイブ利用
  3. 学習済＋学習中アーカイブ利用

カテゴリ 1 のパラメータ $\theta$ と $d\gamma$ については，第 4 章で得た知見より，二番目は大きすぎず小さすぎない値（$\theta = 0.6, d\gamma = 0.6$）を，三番目は小さい $\theta$ と大きい $d\gamma$ の値（$\theta = 0.0, d\gamma = 1.05$）を採用した．

### 7.2.2 評価指標とパラメータ設定

実験は各ケースでシードを変えて 10 試行を行い，最終的にエージェントが報酬 $(16, 16)$ の組を得る政策を獲得できたかどうかを評価した．例題の設計では，報酬 $(16, 16)$ の組は各々のエージェントが自身の報酬を最大化しようとすると陥る均衡点である．具体的に

は，学習済アーカイブを利用する方法を適用する場合は $MAX\_EPISODE$ 回の繰り返しの間に学習済アーカイブに $(16, 16)$ の組を得る政策が保存されているか，そうでない場合は $MAX\_EPISODE$ 回目のエピソードでエージェントが獲得した報酬を評価する．表 7.1 に実験パラメータを示す．$MAX\_STEP$ はエージェントが一向に目標状態に辿り着かない場合に学習を打ち切るステップ数である．

表 7.1　実験パラメータ

| | |
|---|---|
| $\theta_{learned}$ | 0.5 |
| $MAX\_STEP$ | 100 |
| $MAX\_EPISODE$ | 30000 |
| $\alpha_0$ | 0.1 |
| $\gamma_0$ | 0.95 |

### 7.2.3　実験結果

図 7.3 に全ケースの結果を示す．この実験環境と設定は第 6 章の実験と同じものを扱っているため，カテゴリ 3 の手法のみ適用しているケース No.01，02，03 は前章で示したものと同様になっている．No.03 の学習中アーカイブも利用するエージェントは学習完了までに 30,000 エピソードを超えていたシードが存在していたことを思い出すと，この 90% という結果はそれが反映されたものであることがわかる．ほとんどの組み合わせでは性能が維持されているか大きく崩れているわけではないことがわかる一方で，いくつかの組み合わせでは全く学習ができないという極端な結果が示されていることがる．

### 7.2.4　考察

#### 学習の性能

前節で述べた通り，この実験環境は第 6 章の実験と同じであることから，カテゴリ 3 の手法に加えて他のカテゴリの手法を適用しても，同等の性能を有していることが望ましい．次に示す二つのケースを除けば，大きく性能が崩れるわけではないことが明らかになった．具体的には，ケース No.20 と 27 がカテゴリ 3 の手法に加えて他のカテゴリの手法を適用するケースの中でも特に性能が崩れている．この二つのケースの結果は興味深い．どちらもカテゴリ 1 の割引率を増加する方法が適用されているが，カテゴリ 2 を適用しない場合に学習済アーカイブを利用すると報酬 $(16, 16)$ を獲得できなくなってしまう．一方で，時限増加平均差分内部報酬を利用すればその問題は解決される代わりに，学習済＋学習中アーカイブを利用する場合に報酬 $(16, 16)$ を獲得できなくなる．

| カテゴリ1<br>割引率変更 | カテゴリ2<br>内部報酬 | カテゴリ3<br>アーカイブ | (16,16)の獲得[%] | Case |
|---|---|---|---|---|
| 変更なし | なし | なし | 0 | No.01 |
| | | 学習済 | 100 | No.02 |
| | | 学習済＋学習中 | 90 | No.03 |
| | 平均差分 | なし | 90 | No.04 |
| | | 学習済 | 90 | No.05 |
| | | 学習済＋学習中 | 100 | No.06 |
| | 時限増加平均差分 | なし | 0 | No.07 |
| | | 学習済 | 80 | No.08 |
| | | 学習済＋学習中 | 100 | No.09 |
| γ減<br>θ＝0.6<br>dγ＝0.6 | なし | なし | 0 | No.10 |
| | | 学習済 | 100 | No.11 |
| | | 学習済＋学習中 | 80 | No.12 |
| | 平均差分 | なし | 100 | No.13 |
| | | 学習済 | 90 | No.14 |
| | | 学習済＋学習中 | 100 | No.15 |
| | 時限増加平均差分 | なし | 0 | No.16 |
| | | 学習済 | 90 | No.17 |
| | | 学習済＋学習中 | 90 | No.18 |
| γ増<br>θ＝0.0<br>dγ＝1.05 | なし | なし | 0 | No.19 |
| | | 学習済 | 0 | No.20 |
| | | 学習済＋学習中 | 100 | No.21 |
| | 平均差分 | なし | 100 | No.22 |
| | | 学習済 | 80 | No.23 |
| | | 学習済＋学習中 | 100 | No.24 |
| | 時限増加平均差分 | なし | 0 | No.25 |
| | | 学習済 | 90 | No.26 |
| | | 学習済＋学習中 | 0 | No.27 |

図 7.3　報酬 (16,16) を獲得した割合 (10 試行)

学習済アーカイブだけを利用する方法に関してみると，本来安定していた学習（第 6 章の結果を参照）がカテゴリ 2 の内部報酬を適用することによって安定が崩されていることがわかる．一方で，学習済＋学習中アーカイブを利用する方法に関しては，カテゴリ 2 の平均差分内部報酬との相性が特に良く，その二種の組み合わせ（ケース No.06，15，24）では全て 100% という結果が示されている．これはカテゴリ 2 の平均差分内部報酬の「相対的に低い報酬値の目標状態へは向かわない」という強いバイアスによって，学習中のアーカイブを利用する場合に稀に陥る問題（報酬の低い非パレートである政策に対して完了と判断されない程度に学習を進めた後にアーカイブから削除されても，高い確率でこの報酬に向かうため，学習が停滞すること）に陥りにくくなるためと考えられる．

ケース No.04 の結果より，カテゴリ 2 の平均差分内部報酬はアーカイブ利用することなしに報酬 $(16, 16)$ を獲得できる能力があることも明らかになった．

ケース No.20 はカテゴリ 1 の学習が遅れているエージェントの割引率を上げる手法の

みを適用した結果を示しており，この手法は獲得したゴール報酬を周辺の状態-行動価値へ万遍なく割り当てるように推定することによって行動選択の偏りを抑えて探索範囲を広げるものであることから，政策が学習完了と判断されるまで膨大な時間がかかることが，学習済アーカイブの働きを妨げた要因となったと考えられる．実際，このケースでは報酬 $(4, 4)$ 以外の組が獲得されていることは確認されている．ケース No.23 と No.26 の結果は，これに対して，低い報酬の獲得を避けることができるカテゴリ 2 の内部報酬が適用されることで報酬 $(16, 16)$ が獲得できるようになったことを示している．同様に，周辺の探索さえされれば，学習中のアーカイブも低い報酬の獲得を抑制することができるため，ケース No.21 に示すように報酬 $(16, 16)$ の獲得が促されたと言える．

ケース No.27 は分析を詳しく進めると，報酬 $(17, 11)$，$(11, 17)$ の政策を学習済みとした後，報酬 $(18, 6)$ あるいは $(6, 18)$ への学習を進めるが，報酬 6 へ向かうエージェントのエントロピーが下がらないことが報酬 $(16, 16)$ の獲得を妨げているという事実と報酬 $(12, 12)$ が学習済みにならない事実が確認できた．後者は報酬 $(12, 12)$ が学習済みとなる以前に報酬 $(16, 16)$ の学習中アーカイブに保存される（確率的にではあるが十分な可能性で起こる）ことによって報酬 $(12, 12)$ への学習が抑制されるようになるためである．さらに，それまでに進められた学習によって高められた報酬 $(12, 12)$ へ向かうような状態-行動価値は，報酬 $(18, 6)$ へ向かうものと拮抗するようになるため，前者の事実のように報酬 6 へ向かうエージェントのエントロピーが下がらないことが引き起こされたと考えられる．また，カテゴリ 2 で適用されるのが平均差分内部報酬であれば，いずれ報酬 $(16, 16)$ への経験を積み重ねてそのような政策を学習済みと判断されるまでになるまで学習を進めることができるが，時限増加平均差分内部報酬は学習の初期以外では報酬の高い目標状態を目指す力が弱まることも要因の一つである．以上のことから，ケース No.27 の結果は各々の手法の特徴が悪いほうへ積み重なってもたらされたと言える．

**獲得できる政策**

カテゴリ 2 の平均差分内部報酬は全体として優れた性能を示したが，一方でパレート政策のアーカイブを利用するカテゴリ 3 の手法を適用しても $(16, 16)$，$(11, 17)$，$(17, 11)$ を得るための政策は獲得された（学習済アーカイブに保存された）がパレートであるにもかかわらず $(6, 18)$，$(18, 6)$ については獲得されかなった．この理由として，高い報酬を探索しようとする平均差分内部報酬が大きく影響したことが考えられる．パレート政策探索手法はエージェントが得た政策が非パレートであるとわかるまで全ての学習を許すだけの手法であるため，エージェントが全てのパレート報酬を獲得できない場合，当然そのような政策を獲得することができない．平均差分内部報酬は平均より低い報酬へ向かうことを明示的に避けてしまうため，学習が進められることのない学習完了した目標状態へ消去法的に向かうことになり，学習ができるところから順番に学習させてしまおうというアーカ

イブ利用の思想と競合が生じていることになる．別の問題として，「パレート政策を全て獲得させる」というモチベーションからは外れた結果が得られたとも言える．

以上のことから，手法の統合にはいくつかの注意事項が必要となることは明らかである．具体的には，(1) 個々手法の段階でより厳密な分析を行う必要があることは当然のことであるが，(2) 学習メカニズムの同士の組み合わせに際して，組み合わせられるものとそうでないもの，組み合わせて良いクラスと悪いクラス，これらを分け隔てる要素は何であるのか探ることで，統合によって生じる問題を未然に防ぐことが重要になると考えられる．

# 第 8 章

# モデルに内在する問題の体系化

本章では，第 4 章，第 5 章，第 6 章において個別の問題に対処してきた結果を受けて，まとめとしてマルチエージェント学習のモデルに内在する問題の体系的な整理を試みる．本研究で焦点を当てた強化学習を学習の一手法と捉えて，マルチエージェント強化学習の範囲に囚われずに，複数の主体の関わる問題の解決手法にまで広げて議論する．

## 8.1 モデルに内在する問題の位置づけ

本研究では，マルチエージェント学習を同一環境上で複数の主体が動作する対象問題に適用することで何らかの形で協調方法を提示するための技術として捉えている．エージェントの学習結果はモデルに依存するため，究極的には適切にモデル化することが問題解決の全てであると言える．しかし，問題を正しくモデル化することの難しさは，対象問題を解くことのそれと変わらないと言ってよいほど困難である．その中でも，少しずつでも適切なモデル化の実現に近づけるために，対象問題のモデル化という視点まで戻って整理することが重要であると考える．一つのエージェントシステムの設計手順は，(1) 問題の選択，(2) 設計対象の要求定義，(3) エージェントシステムの設計，(4) エージェントの設計，(5) 実験と試験，(6) 評価と改良という六段階を経て行われるものである [13]．この手順には適用先の決定から，運用までが含まれているが，ある程度の段階までは考えるまでもなく先に与えられる場面の方が多いと言える．本研究の立場は，学習結果の全てはモデル化の段階で決まるというものであるため，上記の手順の (2)〜(4) が論点である．図 8.1 に示すように，マルチエージェント学習のモデル化では，(i) エージェントに何を学習させたいのか？，(ii) エージェントに何を頼りに学習させるのか？，(iii) エージェントにどのようなメカニズムをもって学習させるのか？という三つの段階を経て行う必要がある．一つ目については，対象問題を適切な状態空間，行動空間に落とし込めない場合に問題が生じる．この問題の下では，仮にエージェントが神の視点のように全体を完全に把握し，的確に行動できて，かつ正確に学習できたとしても対象問題に対して有益な知見をもたらす

ことができない．これは「認知のモデル」と呼べる．例えば，第 2 章で述べた不完全知覚問題は状態空間の設計が不十分であることが原因であることからこの問題に分類できる．二つ目については，強化学習であれば対象問題の状態や行動空間に対する評価にあたる適切な報酬関数や制約が与えられない場合に問題が生じる．これは「環境のモデル」と呼べる．例えば，報酬分配問題はこの問題に分類できる．三つ目については，適切な学習メカニズムを利用しない場合に問題が生じる．これは「学習のモデル」と呼べる．例えば，同時学習問題はこの問題に分類できる．同時学習問題を考慮していない不適切な学習メカニズムは学習を正しく行うことができない．

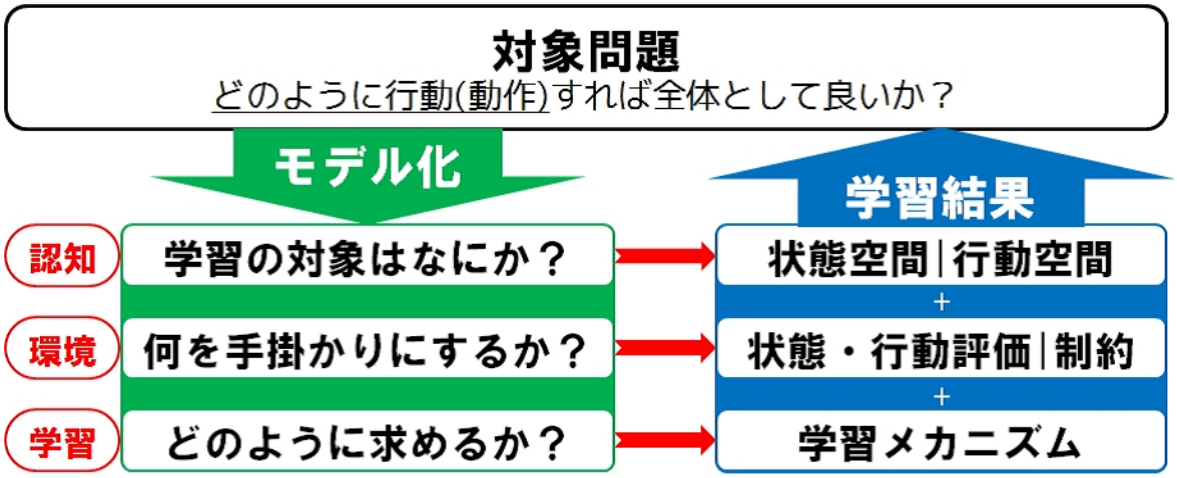



図 8.1　マルチエージェント学習のモデル化の段階によるモデルの分類

これまで，ある学習メカニズムの問題を扱う場合，解決方法として状態空間や報酬関数をチューニングすることも含めて考えれられてきたが，学習メカニズムの問題は「学習のモデル」の中で留めて解決しなければならないし，そもそも問題自体もモデル化の段階によって隔てられているはずである．もし明らかに複数の段階に問題があるとすれば，それぞれに対して総合的にではなく，個別に解決されるべきである．以上のように，問題をモデル化の段階によって分けて考えるという見方がモデルに内在する問題の根本的な立場である．

## 8.2　モデルに内在する問題への対処と展望

本研究は現状で，上で述べた統合的な対処法の探究はほとんど行うことができていない．しかし，Q 学習の観点で言えば，触れられていないコンポーネントは例えば，行動選択手法，step 時間などほとんど残されていない．本研究で対象とした領域に関しては完全ではないものの多くをカバーできたと言える．

ここでは今後，どのうようにすることでモデルに内在する問題全体に対処できるかにつ

いて考える．前節で，モデルに内在する問題はモデル化の段階によって隔てられて存在すると述べた．そして，具体的な対処においては，暗黙にモデル化された部分が焦点であり，改善の余地が残されていた．このことから，モデル化の各段階における既存のモデルについて触れることで，今後どういう方向からモデルに内在する問題へアプローチできるかについて図 8.2 に沿って述べる．第一に「認知のモデル」については，状態空間，行動空間についてであり，既存のモデルは離散空間，連続空間といったように分類される．このモデルでは一般的に詳細にするほど適切なものに近づけると言えるが，状態空間爆発の問題，通信のボトルネックの問題の影響で，それに対処するための様々な形の実装モデルが提案されている．そういった経緯でできた実装モデルには内在する問題があるが，何もかもを解消する必要はなく，マルチエージェント学習では不完全知覚問題を解消するために，モデルの改善が必要になる．第二の「環境のモデル」についても同様で，既存のモデルである報酬関数や教師信号，自然淘汰などに内在する問題について，報酬分配問題を解消するための改善が必要になる．第三の「学習モデル」についても同様で，既存モデルである強化学習や進化的計算，ニューラルネットなどあたりまえとなっているモデルに内在する問題について，同時学習問題を解消するために改善が必要になる．

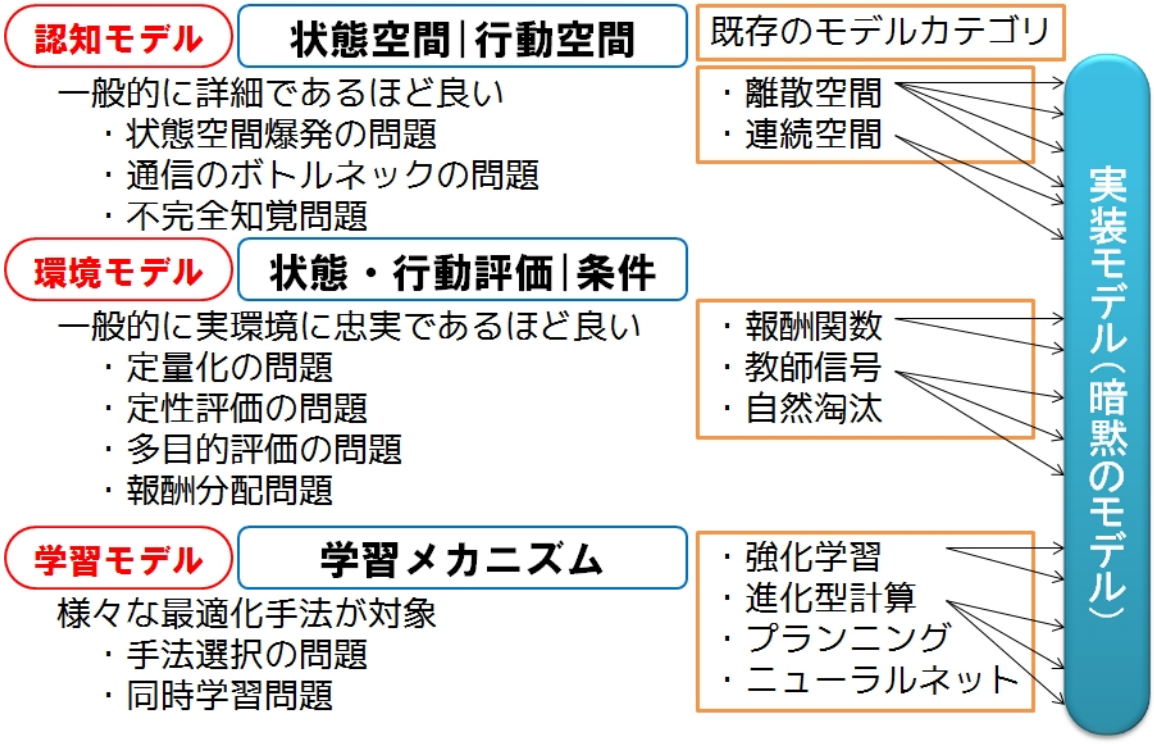



図 8.2　モデル化の各段階における既存のモデルとモデルに内在する問題の関係

モデル化の段階に分けて考えると，全く別の従来法でも同じモデルに対して何らかのモデルが割り当てられていると捉えることができることから，既存の様々な手法はこれまで

の常識に囚われることなく組み合わせられることで，マルチエージェント学習環境において全く新しい成果を生みだす可能性があると考えられる．この場合，様々なモデルの組み合わせ探索手法などの開発ということも考えることができる．

## 8.3 モデルに内在する問題の導出によるアドバンテージ

前節で述べたように，モデルに内在する問題を導出しても結局，表面的には第 2.2.2 章で紹介した三つの問題を解消することとほとんど変わらない．しかし，様々な問題に対して個別に解決するにあたって，好き勝手に各部のモデルを変更するのではなく，他のモデルにどう影響し変化を及ぼすのかを明示的にすることが重要である．一つの問題を解消するために別の問題を引き起こすのでは意味がないため，あるモデルを変更したとき他のどのモデルにどのような変化をもたらすか，あるいは問題を解決するとき無意識に領域を超えてモデルを変えていないかを強く意識しなくてはならない．このように，これまで暗黙に用いられたモデルに対してモデルに内在する問題という観点から捉えることによって，意識的なモデル化を心掛けることで個別の対処が全体として，ひいては複数の手法の統合に際しても適切に機能するモデルが設計できると考えられる．

マルチエージェント学習において無意識なモデル化が顕著なのは他者間で用いられる協調プロトコルのモデルであると考えられる．例えば本論文では，第 4 章と第 6 章にてエージェント間の通信を前提とした学習メカニズムの改善を行った．これらは，学習モデルのみを対象にして部分的な改善を行ったとしながらも，エージェントは他のエージェントとの通信を通して間接的に別の部分のモデルを変更したかもしれない．では，エージェントは通信によって認知モデルを変化させたのか？これに対して，提案エージェントはいずれも依然として初めに与えられた状態空間と行動空間に対応する状態-行動価値を学習していることから，認知モデルには変化はないと言える．次に，エージェントは通信によって報酬や制約を定義する環境モデルを変化させたのか？これに対して，Q 学習の学習パラメータを変更する提案エージェントも学習モデルを変更しているのみで，複数政策を保持するエージェントも報酬や制約を変更していない．しかし実際には学習モデルの内部では変化が起きている．例えば，平均差分内部報酬のエージェントは性質としてエントロピーをそれほど低下させることなく学習を進める．これは「学習が進めばエントロピーが下がる」という一般的な前提と異なっている．このため，この前提は学習完了をエントロピーの低下によって検知しようとする学習済み政策群のアーカイブを利用する方法と齟齬があるため，統合において問題が生じる結果となった．この事態に対して，エントロピーという定量的な変数をモデルとして捉えることで，ある対処が別のモデルに及ぼす影響を明らかにすることで統合時の齟齬は未然に防ぐことができたと考えられる．あるいは，こういった明示的にモデル化されていない指標に依存して構成する方法は，無意識に問題を内

在する可能性を高めるということを明らかにしたといえる．

以上のように，あらゆる要素のモデル化によって現存のあるいはこれから生まれる様々なメカニズムを一元化することは，モデル同士の複雑な相互作用を前提とするマルチエージェント学習の領域において，無意識的なモデル化による問題の発生を抑制することに繋がると期待できる．モデルに内在する問題の導出は，現存の問題をモデル化という視点から明確にするという点でマルチエージェント学習全体の問題のまとめあげの出発点として貢献すると考える．

# 第 9 章

# 結論

## 9.1　本研究の成果

本研究では，Q 学習に代表される強化学習手法をマルチエージェント環境に適用する際に，無意識に行っているモデル化に起因する問題に着目し，その解決手法の提案と有効性の検証を目的とした．これに対して，マルチエージェント学習の問題をモデルに内在する問題という新しい視点から整理した．モデルに内在する問題への具体的な対処として，三例に対して対処方法を検討し，それによって対処の方針を示した．具体的には，(1) 他エージェントとの学習進度の差に基づいて学習速度を調整することによって学習を促進させる手法，(2) 外部の獲得報酬を基に内部報酬を算出し，より高い報酬を探索させる手法，(3) エージェント間のパレート報酬を導く政策を複数保持・利用する手法を提案し，モデルに内在する問題の解決を図る．シミュレーション結果を通して提案手法の有効性を検証したところ，提案手法に基づくエージェントは局所的な政策の獲得に陥らず，大局的に最適な政策を獲得可能であることを示した．

まず，第 1 章では，Q 学習に代表される強化学習手法をマルチエージェント環境に適用することが強力な問題解決能力を期待できる反面，従来から無意識に行っている環境やエージェントのモデル化に起因する問題に着目し，その解決手法の提案と有効性の検証を目的とすることを述べた．

第 2 章では，本研究のベースとなる強化学習を概説し，その後にマルチエージェント環境への適用とその問題点を述べ，最後に本研究の位置づけを明確にした．強化学習のモデルを構成する要素として環境とエージェントの二つのモデルに分けることで，マルチエージェント強化学習との差異を明確に示した．マルチエージェント強化学習では正しく設計されたモデルでもエージェントの学習を含めた複雑な相互作用によって局所的な政策を獲得する問題をモデルに内在する問題と称して，ありふれた設計に無意識に潜む重大な問題であることを指摘した．

第 3 章では，本研究で対処する三つのモデルに内在する問題について詳しく述べ，それ

を含む例題を提示した．一つ目の一定の学習速度のモデルに内在する問題は，エージェント間の学習進度に差が生まれることによって学習の阻害が起こり，場合によっては全く学習が進められない状況を引き起こす．二つ目の外部報酬の受容のモデルに内在する問題は，エージェントが環境から与えられる報酬をそのまま受け取り価値の更新に利用することによって，他に高い報酬があるにもかかわらず局所的な低い報酬に向かうような政策の獲得を引き起こす．三つ目の単一の政策のみの学習のモデルに内在する問題は，エージェントが多数の望ましくない報酬に対して学習を行うことで本当に学習したい大局的な最適解が学得できない状況を引き起こす．それぞれの問題を扱うための例題として，狭路すれ違い問題，マルチステップタスク割り当て問題，マルチステップ 4 タスク問題を提示し，各モデルが引き起こす問題に対処するためのエージェントの構築に対する準備を行った．狭路すれ違い問題は，一方のエージェントのみが学習を進め，他方のエージェントが学習を進められない状況が頻繁に引き起こされ，結果として全てのエージェントが目標状態へ到達できる政策を獲得できないことが生じる問題がモデル化されている．マルチステップタスク割り当て問題は，報酬獲得の難易度と最適性が反対になるような環境で，獲得しやすい報酬に対して優先的に学習を進めることで最適な（高い報酬を得る）政策が獲得されにくい問題がモデル化されている．マルチステップ 4 タスク問題は，エージェントが行う可能性のあるタスク（目標状態）が複数ありそれぞれに様々な報酬値が設定されていることから，それらに対する余計な学習が最適な報酬に対する学習を阻害する問題がモデル化されている．

第 4 章では，一定の学習速度のモデルに内在する問題に対して，学習進度の違いがエージェント間の協調に影響を与えるが，通信を介して共有した学習進度を基に学習速度を調整する提案手法によって，エージェントの競合を回避できることを示した．特に，狭路すれ違い問題に提案手法を適用し，シミュレーション結果を通してその有効性を検証し結果，(i) 学習が進んでいるエージェントの割引率 $\gamma$ を下げる方法は，価値が高く選択されやすい行動価値を重点的に下げ，政策の偏りを防ぐことで，全てのエージェントが目標達成のために学習できる機会を増加させる働きがあること，(ii) 学習が遅れているエージェントの割引率 $\gamma$ を上げる方法は，目標達成につながる行動とつながらない行動の価値をはっきり分けるように推定するため，報酬獲得の機会が少ない状況から効率よく学習する働きがあることを明らかにした．

第 5 章では，外部報酬の受容モデルに内在する問題に対して，複数の報酬に対する報酬獲得の難易度の違いから局所的な政策に陥り易いが，外部報酬を基に見積もった内部報酬を用いて状態-行動価値を更新する手法によって，低い報酬へ向かう政策の獲得を避け，高い報酬へ向かう政策を獲得できることを示した．特に，マルチステップタスク割り当て問題に提案手法を適用し，シミュレーション結果を通してその有効性を検証し結果，(i) 高い報酬を集中的に探索するためには，今までに獲得した報酬の平均値を基準にして外部報

酬を評価し直した内部報酬が有効であり，(ii) この内部報酬が最短経路の探索にも貢献することを見出した．

第 6 章では，単一政策の保持のモデルに内在する問題に対して，多数の望ましくない報酬が望ましい報酬に対する学習を阻害するため，学習途中で見つけた有望なパレート政策をアーカイブ保存し，それに基づいて状態-行動価値の更新を決定する提案手法によって，局所政策の獲得を回避できることを示した．特に，マルチステップ 4 タスク問題に提案手法を適用し，シミュレーション結果を通してその有効性を検証し結果，有望なパレート政策をアーカイブ保存することで，高い報酬の獲得の可能性を維持し，低い報酬への学習を抑制することができることを示すことに成功した．

第 7 章では，モデルに内在する問題への個別の対処方法が複合的なモデルに内在する問題を解決できるかどうかという疑問に対して，ここまでの三つの提案手法を実際に統合し，例題へ適用することでその調査を行った．具体的には，マルチステップ 4 タスク問題に統合手法を適用し，シミュレーション結果を通してその有効性を検証し結果，多数の組み合わせでは他の解決法の効果を邪魔しないという結果が得られた一方で，各手法の特徴が重なり合わさることによって特徴的な性能が得られることがあることを示した．

第 8 章では，前章までに個別の問題に対処してきたことを受けて，モデルに内在する問題およびマルチエージェント強化学習の問題に対する本研究の知見と立場，現状を整理した．具体的には．マルチエージェント学習の領域におけるモデルに内在する問題と本研究の位置づけを述べ，具体的な対処方法の知見をまとめ，統合的見方によってそれぞれの役割を整理した．最後に，統合に関する調査より，個々手法の段階でエージェントの学習に与える影響をさらに細かく分析する必要があることを明らかにした．

## 9.2 今後の課題

### 9.2.1 認知・環境・学習のモデルを考慮した設計論

マルチエージェント学習におけるモデルに内在する問題へのアプローチ方法は述べたが，この視点から実際に既存手法の分析すること，実際に対処を実施することが課題である．この課題の実施を通して，新しいモデルの構築に際してどういった観点が重要となるかを探り，完全なメカニズムに対する知見を増やすことが課題である．そのために，具体的に下記を行うことが考えられる．

- 個々のモデルの改善手法が統合時どういった影響を与えるのか，影響の種類とその記述に関する研究
- これまでの提案されてきた数々の改良手法の様々な要素のマルチエージェント学習のモデルに対するマッピング

### 9.2.2 提案手法に関する課題

**共通する課題**

各提案手法における共通の課題として，(i) 手法の適用の限界や能力を理論的・定量的な観点から考察すること，(ii) 有効性の範囲を調べるためサイズの大きな環境やエージェントが四体以上の問題での評価実験，そして上で述べた課題のさきがけとして，(iii) 手法が統合時どういった影響を与えるのか，影響の種類とその記述に関して研究することなどが挙げられる．

**学習進度に基づく学習速度の調整手法について**

他の競合問題（例えばゲーム理論的な問題）で評価実験することや，エントロピーを正規化して手法を一般化すること，エージェント間の関係を測る尺度として行動-価値関数のエージェント間相互情報量を用いた改良などが挙げられる．また，非競合問題において適用した場合，提案手法が及ぼす影響を調べることが挙げられる．

**内部報酬に基づく大域的最適解探索手法について**

他エージェントの状態や目標状態までの経路によって獲得報酬が変わる問題クラスへの対処が挙げられる．同じ目標状態では常に一定の報酬であることを前提とした環境と異なり，目標状態への行動の抑制がいつでも有効に働かない環境に対応することは重要である．また，報酬獲得の難易度や報酬数に対する性能の調査が挙げられる．

**パレート報酬を考慮した政策探索手法について**

(1) 過去にアーカイブ保存した任意の政策を呼び出して現状を打破する効果的な扱い方について検討すること，(2) 保存した報酬群を基に新しい政策を生成する方法（例えば，Q テーブルの要素の重ねあわせを考えるなど）を検討することなどが挙げられる．また，様々なパレート報酬の形状に対する性能の調査が挙げられる．

# 謝辞

本論文をまとめるにあたり，ご指導，ご鞭撻を頂きました電気通信大学総合情報学専攻髙玉圭樹教授に心から感謝の意を表します．また，指導教員である吉浦裕教授，論文審査委員である西野哲郎教授，髙橋治久教授，内海彰教授，高橋裕樹准教授には，多くの貴重なご意見・ご指摘を頂きましたことを感謝いたします．研究の進め方や研究者としての考え方についても多くのご助言を頂きました服部聖彦助教授，佐藤寛之助教授に心からお礼申し上げます．最後に，長い学生生活を支えてくれた家族に心より感謝を申し上げます．

# 参考文献

[1] Arai, S. and Ishigaki, Y.: ”Information Theoretic Approach for Measuring Interaction in Multiagent Domain”, Journal of Advanced Computational Intelligence and Intelligent Informatics, Vol.13, No.6, pp.649-657 (2009)

[2] 荒井 幸代 : “マルチエージェント強化学習–実用化に向けての課題・理論・諸技術との融合”，人工知能学会誌，Vol. 17，No. 4，pp. 476-481 (2001)

[3] Arthur, W. B., Holland, H. J. et al.: ”Asset Pricing Under Endogenous Expectation in an Artificial Stock Market”, Santa Fe Institute Working Papers, No.9, pp.6-12-093 (1996)

[4] Benda, M., Jagannathan, V. and Dodhiawala, R.: ”On Optimal Cooperation of Knowleadge Sources: An Empirical Investigation”, Technical Report BCS-G2010-28, Boeing Advanced Technology Center (1986)

[5] Boutilier, C.: ”Planning, Learning and Coordination in Multiagent Decision Processes”, In Proceedings of the Sixth Conference on Theoretical Aspects of Rationality and Knowledge (TARK96), pp.195-210 (1996)

[6] Deb, K., Pratap, A., Agarwal, S. and Meyarivan, T.: ”A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II”, IEEE Transactions on Evolutionary Computation, Vol.6, No.2, pp.182-197 (2002)

[7] Epstein, J. M., Axtell, R.: ”Growing Artificial Societies: Social Science from the Bottom Up”, The MIT Press (1996)

[8] 藤田 肇，石井 信 : “部分観測カードゲームのためのモデル同定型強化学習”，電子情報通信学会論文誌， D-II，Vol.88，No.11，pp.2277-2287 (2005)

[9] Hauwere, Y.-M. D. , Vrancx, P., and Nowe’, A.: ”Learning Multi-agent State Space Representations”, Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems, Vol.1, pp.715-722 (2010)

[10] Hu, J., Wellman, M. P.: ”Nash Q-Learning for General-Sum Stochastic Games”, JOURNAL OF MACHINE LEARNING RESEARCH, Vol.4, pp.1039-1069 (2003)

[11] Ichikawa, Y., Sato, K., Hattori, K., and Takadama, K.: ”Entropy-based Conflict

Avoidance According to Learning Progress in Multi-Agent Q-learning", Proceedings of the IADIS International Conference on Intelligent Systems and Agents 2012 (ISA2012), F057 (2012)

[12] 石田 亨，桑原 和宏：“分散人工知能（１）：協調問題解決”，人工知能学会誌， Vol.7, No.6， pp.945-954 (1992)

[13] 木下 哲男：“エージェントシステムの作り方”，電子情報通信学会 (2001)

[14] Kitano, H., Tadokoro, S. et al.: "RoboCup-Rescue: Search and Rescue in Large-scale Disasters as a Domain for Autonomous Agents Research", Proceedings of IEEE Conference, SMC (1999)

[15] Littman, L. M.: "Markov Games as a Framework for Multi-Agent Reinforcement Learning", In Proceedings of the Eleventh International Conference on Machine Learning, pp.157-163 (1994)

[16] 森山 甲一，沼尾 正行：“環境状況に応じて自己の報酬を操作する学習エージェントの構築”，人工知能学会論文誌，Vol.17，No.6，pp.676-683 (2002)

[17] 大内 東，山本 雅人，川村 秀憲：“マルチエージェントシステムの基礎と応用―複雑系工学の計算パラダイム―”，コロナ社 (2002)

[18] Pareto, V.: "Manual of Political Economy", Kelley Publishers, New York, NJ, USA (1971)

[19] Stone, P. and Veloso, M.: "Multiagent Systems: A Survey from a Machine Learning Perspective", AUTONOMOUS ROBOTS, Vol.8, pp.345-383 (1997)

[20] Sutton, R. S. and Bart, A. G.: "Reinforcement Learning -An Introduction-", The MIT Press (1998)

[21] Tan, M.: "Multiagent Reinforcement Learning: Independent vs. Cooperative Agent", The 10th International Conference on Machine Learning, pp.330-337 (1993)

[22] Watkins, C. J. C. H. and Dayan, P.: "Technical note: Q-learning", Machine Learning, Vol.8, pp.55-58 (1992)

[23] Watkins, C. J. C. H.: "Learning from Delayed Rewards", PhD thesis, University of Cambridge, (1989)

[24] Yanagisawa, N., Kawamura, H., Yamamoto, M. and Ouchi, A.: "Quantification of Interactive Behavior in Multiagent Systems", IEIC Technical Report (Institute of Electronics, Information and Communication Engineers), Vol.103, No.725, pp.71-76, (2004), in Japanese

[25] 柳沢，川村，山本，大内：“マルチエージェントシステムにおける相互作用の定量的分析法に関する基礎研究”，情報処理学会　研究報告，CIS-135, pp.101-106, (2004)

[26] Yang, E. and Gu, D.: "Multiagent Reinforcement Learning for Multi-Robot Systems: A Survey", (2004)

[27] Yoshii, T., Akahane, H. and Kuwahara, Y.: "An Evaluation of Effects of Dynamic Route Guidance on an Urban Expressway Network", Proceedings of the Second World Congress on Intelligent Transport Systems, Vol.4, pp.1995-2000 (1995)

[28] Weiß, G.: "Multiagent Systems: a Modern Approach to Distributed Artificial Intelligence", The MIT Press (1999)

# 関連論文の印刷公表の方法および時期

1. 全著者名：Ichikawa, Y., and Takadama, K.
   論文題目：Designing Internal Reward of Reinforcement Learning Agents in Multi-step Dilemma Problem
   印刷公表の方法および時期：Journal of Computational Intelligence and Intelligent Informatics (JACIII), Vol. 17, No. 6, pp. 926-931, 2013
   (第 5 章に関連)
2. 全著者名：市川 嘉裕，髙玉 圭樹
   論文題目：学習進度に基づくマルチエージェント Q 学習における競合回避
   印刷公表の方法および時期： 計測自動制御学会論文集，Vol. 48，No. 11，pp. 764-772，2012
   (第 4 章に関連)
3. 全著者名：Ichikawa, Y., and Takadama, K.
   論文題目：Designing Internal Reward of Reinforcement Learning Agents for Conflict Avoidance in Multi-step Dilemma Problem
   印刷公表の方法および時期：The 16th Asia Pacific Symposium on Intelligent and Evolutionary Systems (IES 2012), pp. 152-157, 2012
   (第 5 章に関連)
4. 全著者名：Ichikawa, Y., Sato, K., Hattori, K. and Takadama, K.
   論文題目：Entropy-based Conflict Avoidance According to Learning Progress in Multi-Agent Q-learning
   印刷公表の方法および時期：Proceedings of the IADIS International Conference on Intelligent Systems and Agents 2012 (ISA 2012), F057, 2012
   (第 4 章に関連)
5. 全著者名：Ichikawa, Y., Hattori, K. and Takadama, K.
   論文題目：Conflict Avoidance using Information Entropy in Multi-Agent Learn-

ing Environment
印刷公表の方法および時期 : The 14th Asia Pacific Symposium on Intelligent and Evolutionary System (IES 2010), pp. 78-84, 2010
(第 4 章に関連)

6. 全著者名：市川 嘉裕，髙玉 圭樹
論文題目：パレート報酬を考慮した政策群アーカイブに基づくマルチエージェント強化学習
印刷公表の方法および時期： 計測自動制御学会，システム・情報部門，第 41 回知能システムシンポジウム，A22-2，2014
(第 6 章に関連)
7. 全著者名：市川 嘉裕，髙玉 圭樹
論文題目：あたりまえのモデル化から生じる問題に頑健なマルチエージェント強化学習の設計に向けて
印刷公表の方法および時期： 計測自動制御学会，システム・情報部門，第 6 回関係論的システム科学調査研究会，2013
(第 4 章および第 5 章および第 6 章に関連)
8. 全著者名：市川 嘉裕，佐藤 圭二，大谷 雅之，服部 聖彦，髙玉 圭樹
論文題目：マルチステップジレンマ問題における強化学習エージェント間の競合解消に向けた内部報酬設計
印刷公表の方法および時期：計測自動制御学会 システム・情報部門 学術講演会 SSI2012，3E1-6，2012
(第 5 章に関連)
9. 全著者名：市川 嘉裕，服部 聖彦，髙玉 圭樹
論文題目：情報エントロピーを用いたマルチエージェント競合回避手法の提案と分析
印刷公表の方法および時期：合同エージェントワークショップ＆シンポジウム 2010，K-1，2010
(第 4 章に関連)