博士論文

マルチエージェント強化学習における チーミング機構に関する研究

A Study on Teaming Mechanisms in Multiagent Reinforcement Learning

— 連携を促進する組織構造と報酬メカニズムの設計 —

— Designing Organizational Structures and Reward Mechanisms that Promote Collaboration —

2022 年

松波　夏樹

論文内容の要旨

強化学習は近年目覚ましい発展を見せているが，いまだ実用に向けては様々な課題がある．その代表的な例が学習するエージェントが環境中に複数存在するマルチエージェント環境の考慮である．

本研究の目的は，動的及び競争的環境への適用を念頭に，様々な環境複雑性を有する問題を対象に，マルチエージェント強化学習 (Multiagent reinforcement learning ; MARL) によって望ましいチームワークを実現するエージェントを得る方法を明らかにすることである．学習エージェントによるチームワーク実現のために，本研究では大きく分けて次の 2 点の提案を行う．

提案 1　チームを構成するエージェントの組織構造と学習方式に対する工夫
提案 2　報酬メカニズムの設計

まず提案 1 について，MARL による解決を困難とする要因である連続空間，部分観測情報環境，競争的環境といった条件について整理し，これらの特徴を全て持つ過酷な環境での追跡問題を対象に問題設定を行う．そのうえで，環境困難性を緩和するチームの組織構造と，学習方式による対処方法として具体的には次の 2 点の提案を行う．1 つ目は，チームにおいて能力に優れたものが Leader となって他のエージェントに指示を行うLeader-Follower モデルの導入と，Leader から Follower に対する一定の強制力を持った通信を付与することであり，2 つ目は，学習の初期段階において競争的環境にある一方のチームに対しもう一方のチームが「あえて負ける」行動を行うカリキュラム学習と，Train and evaluation による学習フレームワークの工夫である．提案手法の有効性を確認するため，追跡問題に対する学習を複数の手法を用いて実験し，従来からある他の手法と比較して提案手法の有効性を確認するとともに，提案手法を構成する各要素毎の影響についても分析する．

提案 2 は，強化学習においてエージェントの方策を特徴づける報酬設計について議論する．複数のエージェントが同時に学習する MARL では，エージェントの自律性を損なうことなく分権的に，全体として好ましい協調を実現することが望まれる．複数のエージェントが行った共同行動の結果に対する報酬だけではなく，個々のエージェントの貢献度に応じた報酬信号を設計することができれば，学習エージェントは容易に全体にとって望ましい行

動を行うような学習を実現することができる．しかし，協調タスクにおけるインセンティブとしての報酬設計はエージェントの貢献度合に応じて，成果の分配を決める貢献度分配問題(credit assignment problem) に帰着し，協調すべきエージェント達の誘因を損なうことなくシステムの要求目標を実現するような報酬関数を設計することは容易ではない．一方メカニズムデザインでは，ミクロ経済学とゲーム理論の一分野であり，複数の利己的なエージェントをいかにして効率よく取りまとめるかという問題を扱い，社会的余剰が最大となるような設計を行う．本研究では，メカニズムデザインの一例として Vickrey-Clarke-Groves (VCG) メカニズムによる支払いのルールである迷惑料の考え方に基づいて，個々のエージェントが仮に存在しなかった場合の社会の効用の差分に基づいてそのエージェントに対する報酬を計算する手法を提案し，実験を行って評価した．

VCG メカニズムでは，評価対象のエージェントの貢献を評価するために，そのエージェントが存在しなかった場合の外部性を評価する必要がある．VCG メカニズムに基づく支払いによる報酬設計の適用可能性を拡大させるため，エージェントの不在性評価が容易ではない問題であっても適用可能な方法として，評価対象のエージェントが存在しない仮想環境を用いた MARL 手法についても提案する．2 種類の問題設定を対象に学習を行って結果を評価する実験を行い，各種従来手法と比較して議論する．

以上から，本研究では 1. チームを構成するエージェントの組織構造と学習方式に対する工夫及び 2. 報酬メカニズムの設計という大きく 2 点の提案を行い，従来課題であった環境複雑性による学習困難性の緩和と，学習エージェント間のインセンティブ設計を反映した報酬設計による協調の発露について実例を示す．

本研究の成果は，人間がそう遠くない将来に直面する，学習によって駆動する多数のエージェントと共生する社会，すなわち AI エージェント同士，あるいは人エージェント同士，さらには人及び AI エージェントが混然一体となった社会状況において，互いに望ましいチームワークを実現するための中央集権性と分権性（Centralized/De-centralized）のあり方について，組織構造と報酬メカニズムの設計という側面からの新たな知見と，今後の可能性を示している．

abstract

Reinforcement learning has made remarkable progress in recent years, however, there are still many issues to be solved towards practical use. A typical example is the consideration of Multiagent environments.

This study aims to clarify how MARL can be used to obtain learned agents that can achieve desirable teamwork for problems with various environmental complexities, with applications to dynamic and competitive environments. In order to build teamwork between learning agents, following two major points are proposed.

1. The organizational structure of the team and the learning framework
2. Design of reward mechanisms

For No.1, we summarize the factors that make it difficult to solve the problem by MARL, such as continuous space, partially observable domains, and competitive environment. Then, we set up the pursuit problem with all of these characteristics. We then propose an organizational structure of the team and the learning framework that can mitigate environmental difficulties. The first is to introduce a leader-follower model that employs leader's instruction and coercion, and the second is to introduce a curriculum learning in which the other team "dares to lose" to one team in a competitive environment in the early stage of learning, and to devise a learning framework based on train and evaluation. In order to confirm the effectiveness of the proposed method, we experiment with several methods and compare the effectiveness of the proposed method with other existing methods.

For No.2, we discuss the design of rewards that characterize agents' policy in reinforcement learning. In MARL, where multiple agents learn simultaneously, it is desirable to achieve overall cooperation in a decentralized manner without compromising the autonomy of an each agent. If we can design reward signals according to the contribution of individual agents, rather than just rewarding the results of joint actions taken by multiple agents, the learning agent can easily learn to perform actions that are desirable for the

whole. However, the design of rewards as incentives in cooperative tasks comes down to the credit assignment problem, and it is not easy to design a reward function that successfully incentivize agents. In mechanism design, there is a lot of literature on cooperation and consideration of other agents from an economics perspective. The Vickrey-Clarke-Groves (VCG) mechanism is a well-known payment mechanism, and we borrow its idea for evaluating each agent's contribution to the whole. The payments in VCG measure an agent's contribution by the difference in the sums of values determined by other agents for when a target agent does or does not exist. We evaluate in two scenarios and clarify that it can increase the social utility.

In the VCG mechanism, in order to evaluate the contribution of the agent, it is necessary to assume the absence of that agent. To apply this scheme in reward design of MARL, the question arises as to how we can assume that a target agent does not exist. In order to expand the applicability of VCG mechanism-based payment to reward design, we also propose a MARL method using a virtual environment where the agent to be evaluated does not exist, as a method that can be applied even to problems where the evaluation of the agent's absence is not easy. To confirm the effectiveness of the proposed method, experiments are conducted to train agents and evaluate the results for two different scenarios and the results are compared and discussed with various existing methods.

In this study, we proposed two major points: 1) The organizational structure of the team and the learning framework, and 2) Design of reward mechanisms. We show concrete examples of the mitigation of learning difficulties due to complexity, and the emergence of cooperation through reward design that reflects the incentive design among learning agents.

The results of this study provide important implications on the team structure such as centralized and decentralized, for the society that humans will face in the near future – living together with a large number of agents driven by learning. In that society, it is desirable to achieve teamwork between human-human agents, AI-AI agents, and humans-AIs altogether.

目次

図目次

表目次

第 1 章

序論

1.1 研究の背景

人工知能におけるエージェントとは，センサによって環境を認識し，アクチュエータによってその環境に対して何らかの行動を行ってインタラクションするものをいう [2]．現実の世界では，環境中に独立したエージェントが一つだけ存在するという状況は考えにくい．ほとんどの現実世界はマルチエージェント環境であり，エージェントは自分と同じように独立して環境を認識して行動する他のエージェントを考慮する必要がある．しかし，マルチエージェント環境では考えうる状態数が著しく増大するため，他のエージェントに対する自然な考慮を事前に規定しておく，プリ・プログラムで実現することは複雑性の観点から困難である [3].

多くの複雑な問題領域では，高い性能を有するエージェントを導出する方法として強化学習に対する期待が高まっている．深層強化学習の登場により，強化学習は近年目覚ましい発展を見せている [4, 5, 6] が，実用に向けては様々な課題がある．多くの現実的な問題領域では，マルチエージェント環境であることに加え，複数のエージェントが繰り返し行動を行う，動的に変化する環境であるとともに，競争的な環境であることが普通であるが，いずれの要素も爆発的に複雑性を増大させるため，ほとんどのマルチエージェント強化学習（Multiagent Reinforcement Learning : MARL）アルゴリズムは対処することが困難である [3].

本研究は，動的及び競争的環境への適用を念頭に，様々な環境複雑性を有する問題を対象に，マルチエージェント強化学習によって望ましいチームワークを実現するエージェントを得る方法を追求する．

本論文は，大きく分けて 2 点の提案で構成する．

まず 1 点目として，エージェントの組織構造及び学習フレームワークを工夫することで対処を試みる提案を行う．スポーツや狩猟，及び警備行動など現実世界で同一の目的・目標に従って行動するチームでは，能力の優れた者がリーダーとなってチームメイトに端的な指示

を送ることで統率の取れたチームプレイを可能としている．そのため，マルチエージェント強化学習のような学習エージェントにおいても同様の組織構造が有効であることを示す．具体的には，Leader-Follower モデルというチームの組織構造を導入することで対処を試みる提案を行う．さらに，カリキュラム学習と，学習過程で得られる最良のモデルを保存するためのTrain and evaluation を行うように深層強化学習のフレームワークを工夫することで，より良い結果を導くことについても述べる．カリキュラム学習では，学習の初期段階において競争的環境にある一方のチームに対し，いわば勝ち方を教示するために，もう一方のチームが「あえて負ける」行動を行うカリキュラムを組み込むことが有効であることを示す．さらに，Train and evaluation によって学習中に得られた学習済みモデルの性能を定期的にベンチマークし最良のモデルを残すことで，競争的環境のような不安定な学習環境であっても得られるモデルの性能が向上することを示す．これらの有効性を確認するために，動的及び競争的なマルチエージェント環境であることに加え，連続空間及び部分観測という特徴を持った過酷な環境での追跡問題を対象に実験を行い，有効性と残された課題について述べる．

2 点目は，強化学習においてエージェントに対して最も重要な教示信号となる報酬に対して，公共的な意思決定を自律的かつ分権的に実現するための制度設計であるメカニズムデザインの考え方を取り入れた新たな MARL の報酬設計法を提案し，その有効性を評価する．複数のエージェントが同時に方策を更新していく MARL では，考慮すべき状態数が爆発的に増加し，安定的に学習することが困難となる [7, 8]．しかし，複数のエージェントが同時に学習する場合でも，共同行動の結果に対する個々の学習エージェントの貢献度に応じた個別の報酬をフィードバックとして与えることができれば，各エージェントの方策更新が安定し学習の効率は向上するはずである．ところが，こうした報酬の設計は貢献度分配問題に帰着し [7]，汎用的な設計は容易ではない．これらの課題を踏まえて，本論文では競争的環境において一方のチーム内の協調に焦点を当てることで，新たな MARL の報酬設計法を提案する．メカニズムデザインの分野では，経済学の観点からエージェント間における協調や，他者を考慮するアルゴリズムが提案されている [9]．メカニズムは利己的なエージェントが組織としての意思決定をするためのルールやプロトコルであり，これらを理論的に示しているのがメカニズムデザインである．本研究では，個々の学習エージェントがチームにとって望ましい行動方策を獲得することを誘因するために，それぞれのエージェントの報酬関数において本来の報酬に加え，社会的に効率的な資源配分を実現するメカニズムとして知られる， Vickrey-Clarke-Groves (VCG) メカニズムに基づくアルゴリズムで定義した迷惑料を課す方法を提案する．提案手法を評価するために，個人の利益と全体の利益が競合しうる状況において協調行動が必要となるマルチエージェントタスクを想定した問題を設定し，実験を行って提案手法が協調的方策獲得に対する有効性を分析し，議論する．

1.2 研究の目的

本研究の目的は，1.1節で述べた，動的及び競争的環境においてエージェントの分権的性質を損なわないマルチエージェント深層強化学習の有効な手法を確立することである．学習によってエージェントの行動を獲得することの目的として，主に以下の 2 点が挙げられる．

- エージェントが活動する環境に未知の要素があり，エージェントが遭遇する全ての状況を設計者が予め予想することができない状態空間への対応 [2][3]
- 望ましい結果を規定した場合でも，結果への過程解法をプログラムとして実現するのが困難な問題への対応 [2]

これらの問題の解決を目指した強化学習は，エージェントそれぞれの環境で行う試行錯誤を通じて得られる報酬の情報を基に，環境に適応するエージェントの行動方策を獲得する機械学習の一手法であり，近年は価値や方策の関数近似に多層ニューラルネットを応用した深層強化学習が盛んに取り組まれている．しかし，自律性を有する多数のエージェントが同時に学習を行う環境では，学習の安定性や計算量等，単一エージェントにおける学習とは異なる課題が存在する．さらに，現実の世界はマルチエージェント環境であることに加え，連続空間や，部分観測及び競争的な環境であることが普通であるが，それぞれ考え得る状態数が著しく増加する要因となるため，ほとんどのマルチエージェント強化学習アルゴリズムは計算量等の問題により，対処することが困難である．そのため，本論文では現実適用を見据えた困難性の高い環境を想定したマルチエージェント強化学習において，複数のエージェントが一つのチームとして目的に指向して行動するようなエージェントの一群を得るための提案を行い，実験によってその有効性を確認する．

1.3 本論文の構成

本論文の構成を以下に示す．

まず，2 章で本研究において対象とする問題領域とその解決策となりうる関連研究について述べる．特に，本研究のコアであるマルチエージェント強化学習に関する課題を (1) 部分観測情報環境と競争的環境という強い制約の克服と，(2) 学習エージェントに協調のインセンティブを与えるための報酬設計法の 2 点として整理し，それらの課題に対しどのような提案を行うのかについて述べる．次に，3 章では 2 章で述べた課題のうち，(1) 「部分観測情報環境と競争的環境」という制約をエージェントの組織構造と学習フレームワークを工夫することで対処を試みる提案を行い，その有効性を実験によって評価し，議論する．次に，4 章では 2 章で述べた課題のうち，(2)「学習エージェントに協調のインセンティブを

与えるための報酬設計法」について，メカニズムデザインにおける Vickrey-Clarke-Groves (VCG) メカニズムに基づく支払いを報酬設計に応用することを提案する．VCG メカニズムでは，評価対象のエージェントの貢献を評価するためには，そのエージェントが存在しなかった場合の外部性を評価する必要がある．しかし，対象問題によってはこのエージェントの不在性を仮定することが可能な場合とそうでない場合がある．そこで，5 章では VCG メカニズムに基づく支払いによる報酬設計の適用可能性を拡大させるため，エージェントの不在性評価が容易ではない問題に対する適用方法を提案し，それぞれについて実験を行って結果を評価し，議論する．最後に，6 章では本研究のまとめとして，本研究で得られた知見と今後の課題について述べ，マルチエージェント強化学習によってチームワークを実現することの今後の展望についてまとめる．

図1.1に本論文の構成を各章の関連図として示す．

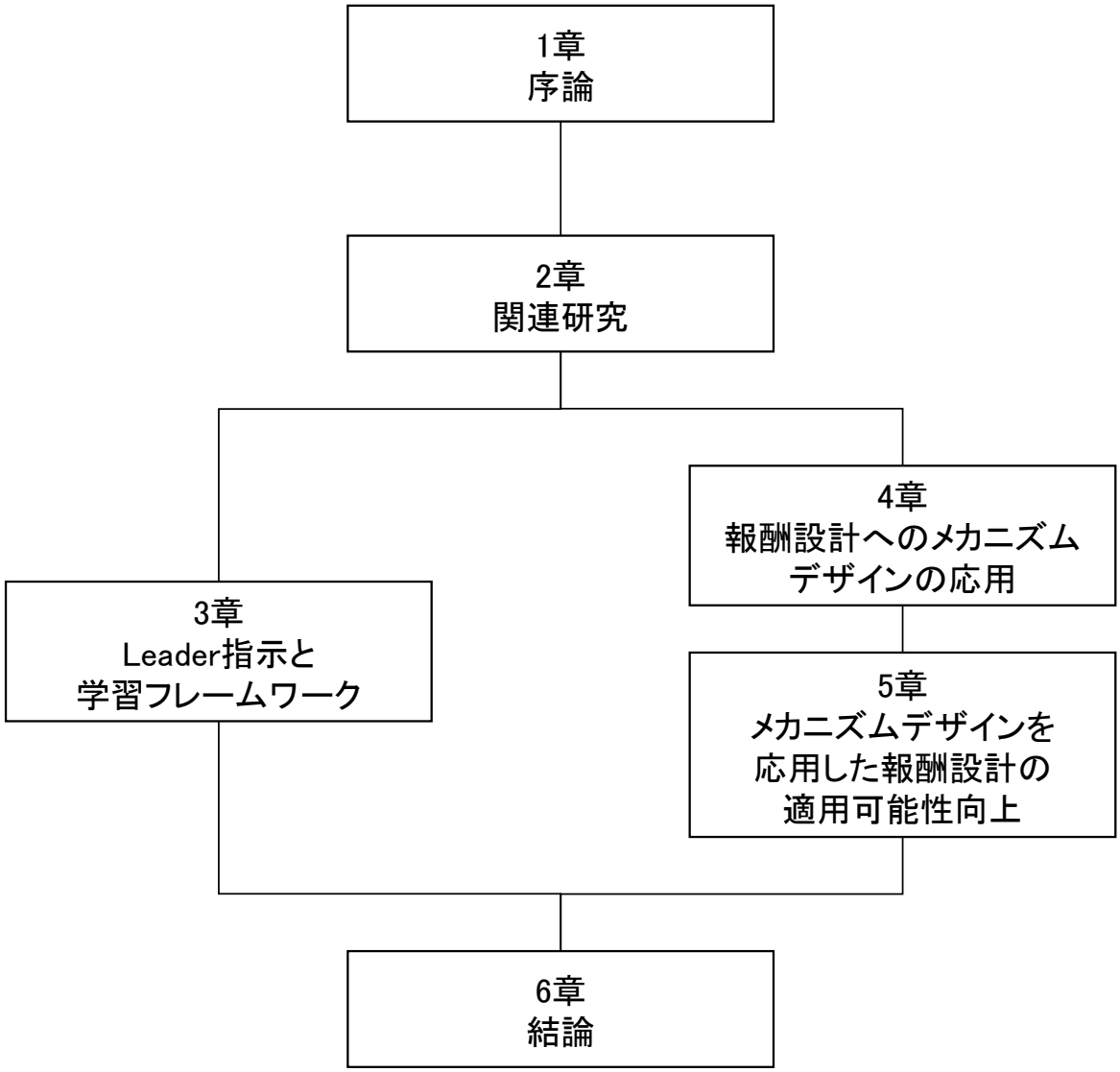


図1.1: 本論文の各章の関連図

第 2 章

関連研究

2.1 序言

本章では，本研究の背景となる関連研究について述べた後，提案内容の概要を述べる．まず2.2節で用語としてのチーム，チームワーク及びチーミングについてまとめる．次に，2.3節で強化学習，2.4節でマルチエージェント強化学習における課題と本研究が対象とする問題領域，すなわち学習によって得られる自律エージェントがチームとして目的に志向して協調する方策を獲得するような問題領域について整理する．次に，2.5節では MARL による解決が困難とな要因の一つである環境の複雑性について整理し，主として3章で取り組む，部分観測情報環境と競争的環境の特徴について述べ，部分観測情報環境における制約を緩和できる可能性のあるエージェント間通信方法として Leader 指示と，競争的環境に対する対処としての学習フレームワークに関する提案の概要を述べる．続いて2.6節では MARL により獲得する方策を特徴づける報酬設計法についての関連研究を示し，2.7節において MARL とは異なる研究領域であるメカニズムデザインの関連研究について示すとともに，MARL の報酬設計へ応用することを提案し，最後に2.8節で本章をまとめる．

2.2 チーム，チームワーク，チーミング

人工知能が議論の対象とする環境におけるエージェントの協調性としての属性について述べるとき，一般に協調性 (Cooperative) と競争性 (Competitive) を軸に分類される [10, 11]．両者は時に混在し，ある 1 対のエージェント間であっても協調的な要素と競争的な要素を同時に持つ場合がある．例えば，タクシー運転手同士は互いに事故を避けながらそれぞれのパフォーマンスを最大化しようとする面においては協調的な要素がある一方，駐車スペースが限られているような場合には競争的な関係となる [2]．一方，本研究で目的とするチーミングの実現について議論するためには，協調性と競争性という軸とは異なる，チームという概念の導入が必要である．

チームとは，例えば Cambridge Academic Content Dictionary によれば “a number of people who act together as a group, either in a sport or in order to achieve something”[12] とあり，チームワークとは，“the combined actions of a group of people working together effectively to achieve a goal”[12] とある．すなわち，チームとは何かを実現するために集まって一緒に行動する一群の人々（グループ）のこと指し，チームワークとはあるひとつの目標を実現するために一群の人々（グループ）が共同行動を取ることを指す．共通の目標を有するグループはチームとなり，チームが一つの目的に指向して行動した場合をチームワークという．一般的な道路交通は信号によって全体が協調的であるが，これはチームワークとは呼ばない．一方，コンボイとして隊列を組んで運転するのはチームワークの一例である．チームワークと呼ぶには，協調だけでなく，少なくともチームの共通目標とチームメンバーによる協力が必要だといえる [13].

それでは望ましいチームワークを実現し，より高い能力を発揮するチームを実現するためにチーム員に求められる特性や組織構造とはどういうものだろうか？例えば強いサッカーチームを作るためには，個人的身体能力がずば抜けた人材を集めるのが手っ取り早い [14]. 同じアナロジーで考えれば，強化学習によって個人効用としての報酬を多く獲得できる学習エージェントを集めれば強いチームが作れるのだろうか．興味深い実験として，鶏の行動に与える遺伝的影響と生産性の関係を調査した例 [15] がある．Muir らによる実験では，産卵鶏を最大限に卵を産むように遺伝的に選抜すると，他の鶏にくちばしでダメージを与える割合が増えるという副作用があった．鶏卵の生産性という観点ではその他にも，繁殖能力が失われてしまったり，ヒステリー，恐怖に対する耐性や食欲など様々な要素によって卵の生産性が左右される．この実験を通じて Muir らは，高い産卵率を確保するためには個別の鳥を選抜するのではなく，ケージ単位などの集団で生産性の高いグループを選ぶことを提案している．この結果を援用して，しばしば人間社会におけるビジネスやスポーツにおいて，能力の高い「スーパーチキン」を集めたチームの生産性が高くなるどころか悪化するという文脈で用いられる [16].

MARL では，個々のエージェントが学習エージェントであり自律性を持つ．自律性を持つエージェントが個人効用のみを追求するような報酬設計によって学習を行えば，貪欲な行動が学習され，チームとしての全体効用が高まるとは限らない．一方，チームとしての全体効用のみを個々のエージェントに対する報酬とすると，自分の努力とは無関係に報酬を得る機会が増え，これもチームとしての全体効用が高まるとは限らない．学習エージェントによるチームとしての能力を高めるためには，個人効用の追求と全体効用への貢献に対するインセンティブが必要といえる．

また，チーミングの定義として，例えば Cambridge ビジネス英語辞典によれば “the

activity of working together as a team”[17] とある．この定義によればチームワークとほとんど違いがないが，実際の使用例では，一つの目的に指向して行動するチームを意図的に構成しようとする場合に用いられるようである [18]．よって本研究では，チーミングを「チームワークを発揮するチームの実現」という意味で用いる．

本研究では，様々な環境複雑性を有する問題を対象に，MARL によって望ましいチームワークを実現するエージェントを得る方法，すなわちチーミング機構を追求する．そのために，環境複雑性を克服するための連携を促進するチームの組織的構造と，個人効用と全体効用への貢献に対するインセンティブを両立するための報酬メカニズムについて検討する．

2.3 強化学習

強化学習は，エージェントが環境中で行う試行錯誤を通じて得られる報酬を頼りに，環境に適応するエージェントの方策を獲得する機械学習の一手法である．事前に教師となるデータを与えるわけではないが，行動の良し悪しを評価するための報酬は事前に設計する．機械が試行錯誤を通じて自らの行動方策を学習するという強化学習の近代的な取り組みは，人工知能の歴史とほぼ同じく 1950 年代に始まり [19]，Q 学習や Sarsa など価値関数を用いた手法が取り組まれてきた．近年，主に画像認識で高い識別能力を発揮した多層ニューラルネットの表現力を用いて，価値や方策の近似に多層ニューラルネットを応用した，深層強化学習が盛んに取り組まれている．離散的な行動を扱う Deep Q-Learning (DQN) [4] や連続量の行動が利用可能な Deep Deterministic Policy Gradient (DDPG) [20] などが基本的な手法として知られており，さらに近年では Rainbow [21] のように様々な工夫を追加した改良に向けた取り組みが数多くなされている．本節では，強化学習において想定するマルコフ決定過程と，基本的な強化学習方式として Q 学習，深層強化学習のうち DQN と DDPG について述べる．

2.3.1 マルコフ決定過程 [19]

強化学習では，環境との相互作用を通じて長期的な目標を達成するように学習するエージェントの重要な側面に着目するため，対象とする問題環境がマルコフ決定過程（Markov Decision Process，MDP）と呼ばれる特徴を持つことを仮定する．環境の状態が直前の状態と，そこでのエージェントの行動のみに依存し，報酬は直前の状態と遷移後の状態に依存するような性質のことをマルコフ性といい，マルコフ性を持つ環境を MDP と呼ぶ．MDP は逐次的意思決定の古典的形式であり，エージェントが行動を行った結果遷移する環境に関する情報と，得られる即時的な報酬をもとに，将来の報酬を最大化する行動を学習するための定式化を容易化する．

MDP では，離散的時刻 $t = 0, 1, 2, \ldots$ においてエージェントが環境状態 S_t を観測し，エージェントが行動 A_t を選択した結果，報酬 R_{t+1} を得るとともに，状態が S_{t+1} に遷移すると仮定する．MDP とエージェントの行動の組み合わせにより，次のような状態，行動，及び報酬という一連の時系列的な遷移が生起する．

$$S_0, A_0, R_1, S_1, A_1, R_2, S_2, A_2, R_3, \ldots \tag{2.1}$$

強化学習の目的は，学習エージェントが目先の即時報酬ではなく，長期的な報酬を最大化するような行動を学習することである．そこで，長期的に得られる報酬として，将来の期待報酬を定義する．MDP の前提のもとで，エピソードが時刻 T で終了する場合の時刻 t における期待報酬 G_t は次のように表される．

$$G_t \doteq R_{t+1} + R_{t+2} + R_{t+3} + \cdots + R_T \tag{2.2}$$

真の期待報酬はエピソード終了後に判明するため，エージェントが行動を決める時点では式(2.2)のような真の期待報酬は明らかではない．そこで，エージェントが行動を決める時点で推定される期待報酬が最大となるような行動を選択するために，G_t を以下のように見積もる．

$$G_t \doteq R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \ldots \gamma^{T-t-1} R_T \tag{2.3}$$

$$= \sum_{k=0}^{T-t-1} \gamma^k R_{t+k+1} \tag{2.4}$$

ここで，γ は期待報酬を見積もる際の不確かさを表現する割引率と呼ばれる係数であり，0 以上 1 未満とする．式(2.3)は，再帰的に次のように表現できる．

$$G_t \doteq R_{t+1} + \gamma \left(R_{t+2} + \gamma R_{t+3} + \cdots + \gamma^{T-t-2} R_T \right) \tag{2.5}$$

$$= R_{t+1} + \gamma G_{t+1} \tag{2.6}$$

強化学習は，式(2.6)で表される推定期待報酬を最大化するような行動を選択する学習エージェントの獲得を行う．

2.3.2 Q 学習

強化学習におけるエージェントの目的は，マルコフ決定過程において将来の割引された期待報酬の合計である推定期待報酬を最大化するような方策 π を見つけることである．Q 学習では，式(2.3)〜式(2.6)で示される推定期待報酬を「価値」と呼び，次のように表現する．

$$v(S,\pi) \doteq \mathbb{E}_\pi[G_t|S_t] \tag{2.7}$$
$$= \mathbb{E}_\pi\left[\sum_{k=0}^{\infty}\gamma^k R_{t+k+1}|S_t\right] \tag{2.8}$$

ここで，$\mathbb{E}_\pi[\cdot]$ はエージェントが方策 π に従って行動する場合に期待される価値，R_t は時間 t における報酬，$\gamma \in [0,1)$ は割引率である．$v(S,\pi)$ は状態 S における方策 π のもとでの価値であり，価値には将来の期待報酬の見積もりが含まれる．つまり，強化学習ではエージェントの適切な価値観を構築し，どのような状態においても将来の期待報酬を最大化するような適切な行動をとるようになることを目的としている．Q 学習では，状態 S における行動 A の価値，すなわち行動価値関数を Q 関数と呼び，次のように定義する．

$$Q^*(S,A) = R(S,A) + \gamma\sum_{S'} p(S'|S,A)v(S',\pi^*) \tag{2.9}$$

ここで，$p(S'|S,A)$ は行動 A を行った時に，状態が S から S' に遷移する確率である．Bellman Equation と呼ばれる Q 関数の更新式は次のように表される．

$$Q_{t+1}(S_t,A_t) \leftarrow (1-\alpha)Q_t(S_t,A_t) + \alpha\left(R_t + \gamma\max_A Q_t(S_{t+1},A)\right) \tag{2.10}$$

ここで，α は学習率であり，即時報酬に基づいてどの程度価値を更新するか調整するためのパラメータである．Q 学習は，時刻 t における現在の行動価値の推定値 Q_t をベースとし，即時報酬 R と割引した次の状態における行動価値に学習率を乗じて更新していくものであることが分かる．また式(2.10)に推定期待報酬や状態遷移確率を含まないことから，環境が未知である場合でも適用可能である．

Q 学習の主な課題は，Q が S と A を引数とすることから分かるように，各状態と行動のセットに対して離散的に Q を定義することから，大規模な問題への適用が難しいというという点にある．例えば囲碁の場合，19 路盤と呼ばれる一般的な碁盤で 19 本の線を持ち，交点の数は 361 となる．それぞれの交点に黒，白，及び石が置かれていないという 3 種類の状態を考慮すると単純計算で $3^{361} = 1.74 \times 10^{172}$ となり，全ての状態を考慮した Q を定義することは現実的ではない．また，状態や行動が連続量を持つような環境では問題を離散化してモデル化するなどの加工が必要となり，簡単ではない．

2.3.3 DQN と DDPG

Mnih らによって 2013 年に提案された DQN [4] は，Q を多層ニューラルネットを使用して表現することで，Q 学習における空間計算量からくる課題の解決を狙った手法である．

DQN では，以下の式(2.11)で与えられる $L_t(\theta_t)$ を，各時刻において最小化するようにニューラルネットの重みに関するパラメータ θ を更新することで，ニューラルネットで表現する $Q(S, A; \theta)$ を学習する．L は loss 関数と呼ばれる．

$$L_t(\theta_t) = \mathbb{E}\left[(y_t - Q(S_t, A_t; \theta))^2\right] \tag{2.11}$$

y_t は時刻 t における更新時に $Q(S_t, A_t; \theta)$ が出力すべきターゲットを表す．L の最適化を計算する間は，パラメータ θ は前の時刻における θ_{t-1} まま固定する．

loss 関数の θ に関する勾配は次の式で与えられる．

$$\nabla_\theta L_t(\theta_t) = \mathbb{E}\left[\left(R + \gamma \max_{A'} Q(S', A'; \theta_{t-1}) - Q(S, A; \theta_t)\right) \nabla_{\theta_t} Q(S, A; \theta_t)\right] \tag{2.12}$$

ここで，Q 学習及びその拡張である DQN は，環境の状態応じた行動の価値を推定して方策を決定するアルゴリズムである．一方，DDPG [20] は方策ベースと呼ばれる方法の一種で，方策 π を学習しようとする方法である．DDPG では，方策の評価と改善の機能を分離し，別々にモデル化する Actor-Critic 法と呼ばれる方法が用いられる．学習を通じて方策改善を行いながら行動を決定する部分を Actor，その方策を評価する部分を Critic と呼ぶ．状態空間及び行動空間が高次元または連続的な空間を扱う場合，方策評価の機能と方策改善の機能を分離した方が有効とされ，さらにそれぞれに対してニューラルネットを用いるのが DDPG である．一般に，基本的な深層強化学習法として知られる DQN が状態空間及び行動空間が離散的である環境を取り扱う一方，連続的である環境を取り扱うのが DDPG である．

2.4 マルチエージェント強化学習（MARL）

2.3節で述べた強化学習法を用いて学習するエージェントは，環境の状態を観測し行動を決定する．この場合，他の学習エージェントが環境中に存在する場合でも，暗黙的に他のエージェントは環境の一部として捉えられる．一方 MARL は，共通の環境を共有し相互作用する自律的な意思決定主体のグループが，環境や他のエージェントとの相互作用を通じて，それぞれが自分の長期的な利益を最適化することを目指す，逐次的な意思決定問題を扱う．マルチエージェントシステムにおける学習は，複数のプレイヤーによるゲームプレイ [22, 23]，マルチロボットの制御 [24, 25]，自動運転車 [26, 27]，通信ネットワーク・ルーティング [28] など，広範囲にわたって研究が行われている．これらの MARL の問題領域は，大きく分けて完全協調型，完全競争型，そして両者の混合型の 3 つに分類される [10, 11]．本研究では一般性を重視するため，各エージェントが利己的であり，報酬が他者の報酬と衝突する可能性があるという混合問題領域 [11] と，エージェント間の情報構造が完全に分散

化された設定に取り組む．本論文では，このような利己的なエージェントによって構成される環境において，目的と一とするチームにおける協調に焦点を当て，様々な環境複雑性を有する問題を対象に，MARL によって望ましいチームワークを実現するエージェントを得る方法を追求する．

現実問題への適用を想定した場合の MARL の課題として，本研究では大きく分けて (1) 環境複雑性と，(2) 貢献度分配問題（credit assignment problem）の 2 点を取り扱う．

まず (1) について，現実の世界は連続空間，部分観測及び競争的な環境であることが普通であるが，これらいずれの要素によっても考え得る状態数が著しく増加するため，ほとんどの MARL アルゴリズムは対処することが困難である [3]．2.3.2節で述べた Q 学習では，状態価値を表として保持するため，連続的な状態量を取り扱うことが容易ではなかった．例えば，Q 学習による追跡問題の MARL 実施例 [29] では，状態空間を 150×150 の 2 次元グリッドとし，その中でエージェントは 8 方向に移動できるような環境を定義している．一方，近年深層学習の隆盛により，学習エージェントが取り扱うことができる状態量や行動空間は顕著に拡大している．特に，近年の MARL における顕著な貢献として挙げられる Multiagent Deep Deterministic Policy Gradient (MADDPG)[1] では，Actor-Critic を用いた強化学習法である DDPG [20] を拡張し，行動価値を推定する Critic には全てのエージェントの行動と観測情報を入力して一元的 (Centralized manner) に学習させ，個々のエージェントの行動を決定する Actor は分散的（Decentralized manner) に強化することで，エージェントの分権的性質を損なうことなく学習及び実行を行うことができる．MADDPG のアプローチの概要を図2.1に示す．この中で例として実験で用いられている追跡問題では，エージェントの状態は位置と速度で与えられ，それぞれ 64 ビット浮動小数点で定義されているため，状態の組み合わせ数は $2^{64^4} = 1.15 \times 10^{77}$ となる．エージェントの行動は 4 方向 + 無効（冗長係数）への 5 つの力ベクトルで与えられ，組み合わせ数は $2^{64^5} = 2.14 \times 10^{96}$ という，連続的な空間を表現するのに十分な状態量と行動の候補を取り扱うことができる．先述の Q 学習による追跡問題の MARL 実施例 [29] における状態の組み合わせ数 $150 \times 150 = 2.25 \times 10^4$ 及び行動の組み合わせ数 8 と比較すると，MADDPG では文字通り桁違いの複雑性を持つ環境を取り扱っていることがわかる．

しかし，MADDPG は競争的環境における協調や連続空間は考慮しているものの，エージェント間の明示的な通信モデルは取り扱っていない [30]．また Critic には全エージェントの観測情報が入力されるため，部分観測情報環境ではエージェント毎の無効な観測情報による影響が全体に波及することが問題として挙げられる．この課題について，2.5節で議論する．

また (2)貢献度分配問題について，強化学習では，学習の過程で得られる報酬に基づいて方策を獲得するため，学習エージェントの方策は，どのような条件でどれだけの報酬を与

えるかによって決定づけられることとなる．従って，複数の学習エージェントが存在するMARL では，個々のエージェントが共同で行動した結果に対する報酬をどのように分割して与えるかという問題が発生する．学習によって行動を変化させるエージェントが同一環境に複数存在するマルチエージェント環境では，あるエージェントにとっての利益は他のエージェントにとっての不利益である状況が考えられる．そのため，考慮すべき状態数が爆発的に増加し，全体として望ましい協調を実現するような個々のエージェントの方策を学習によって獲得することが難しくなる [7, 8]．この課題について，2.6節で議論する．

2.5 部分観測情報環境と競争的環境

2.5節では，MARL による解決を困難とする要因の一つである環境に関する複雑性について整理し，主として3章で取り組む複雑性に関する困難性緩和に資するエージェントの組織的構造やエージェント間通信，学習フレームワークの工夫について論じる．

MARL においてエージェントがインタラクションする環境が有する複雑性によって，

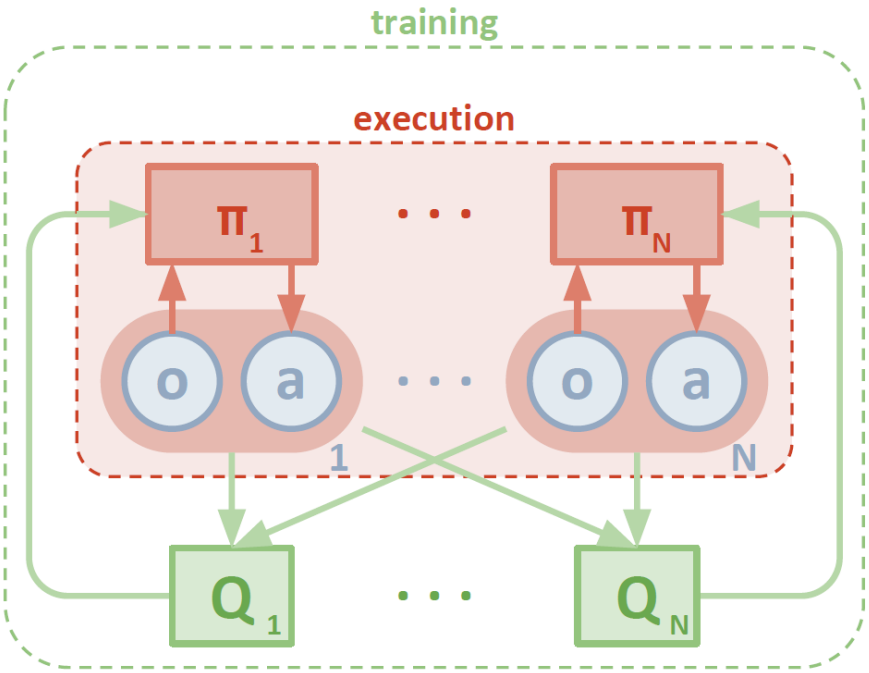


図2.1: MADDPG のアプローチの概要 [1]

解決の困難性は大きく左右される．例えば，エージェント数が増加すると，各エージェント固有の状態と行動に関する変数が全体の状態と行動空間に加算されていくため，MARL の問題複雑性は指数関数的に増加し解決が困難になる．本研究では，MARL を取り扱うためエージェント数は少なくとも 2体以上の問題を取り扱う．

また，2.1節で述べたように，状態量や行動が離散的，連続的であるかによっても MARL の困難性は大きく変わる．深層学習の隆盛と MARL への適用によって近年では連続量を取り扱う例が増加している．本研究では，対象問題によって離散と連続両方の問題を取り扱う．

部分観測情報環境における部分観測マルコフ決定過程 (POMDP) では，エージェントは観測情報として環境の真の状態ではなく不確実性のある情報に基づいて意思決定を行う必要が生じる．不確実性によって計算複雑性が著しく増大し [31]，マルコフ決定過程（MDP）と同様の強化学習手法の単純な適用では対処が難しくなる．過去の時系列的な観測情報を活用することで改善を図る方法として，Deep Q-Network と Long Short Term Memory (LSTM)[32] を組みわせる手法 [33] や，RNN (recurrent neural network) を用いる手法 [34]，Attentional communication と呼ぶエージェント間通信の提案 [35]，など学習機構を工夫する方法などが提案されている．一方，部分観測情報環境において人間がチームプレイを行う場合を考えると，例えばサッカーの場合，声掛けや身振り手振りといったごく限られた情報伝達によって不完全情報を補っているものと思われる．人間が限定的な情報伝達を用いてチームプレイが可能なのは，事前の練習を通じて互いの特性や情報伝達方法を確立しているためと考えられる．また，チームにはリーダーが存在し，チームメイトがリーダーの指示に従うことで統率の取れたチームプレイを可能としている．リーダーは自身だけでなくチーム全体の状況把握に努めるであろうし，全体の状況把握ができる者が選ばれるであろう．Leader-Follower モデルは，サッカーに限らずチームプレイを必要とするスポーツのほか，多人数で行う狩猟や，警備行動などでもみられる．そこで本研究では，3章において追跡問題における Predator チームに Leader-Follower モデルを導入し，Leader 指示を与えることによって不完全情報ゲームにおけるマルチエージェント強化学習でチームワークを向上させることが可能なことを示す．

また，2.4節で述べたように，MARL の問題領域は大きく分けて完全協調型，完全競争型，そして両者の混合型の 3 つに分類される．このうち混合型では，協調と競争が混在する問題領域を取り扱う．3章では，より困難な例として混合型に分類できる追跡問題（Predator-Prey）を取り上げる．追跡問題は，捕食者としての Predator エージェントと被捕食者としての Prey エージェントに分かれ，Predator は Prey を捕獲することを，Prey は Predator に捕獲されないことを目標として同一環境中を移動して競い合うような問題をいう．マルチエージェント強化学習を行う環境中に，追跡問題のように敵対的に行動するエー

ジェントが存在する場合，互いに自身の利得を最大化するために均衡を破ろうとするため，学習の収束性が悪化する．近年では画像生成などを行う敵対的生成ネットワーク [36] の研究が盛んであるが，同様の理由により有意な学習を行うことが難しいといわれる．追跡問題や敵対的生成ネットワークの例のように，敵対的に行動する両者が同時に学習する状況では，競い合う過程で，巧みな行動を行って利得を獲得する相手に対し，さらにそれを上回るような行動を行うような方策を得る， いわば共進化的に両方が高度な方策を学習する場合もある．一方， 双方の成績は相手次第という状況であるため，場合によっては両者がほとんど行動しないような方策を得ることがある．敵対的学習を通じて得られた方策は，学習が収束している場合，学習中に対戦した他方に対する適応戦略になっているため，両者にとっての均衡点が得られているはずである．そのため，ほとんど行動しない等の汎用性のない相手を対象に収束した均衡点の場合，得られる方策もまた汎用性のない方策となる恐れがある．3章では，競争的環境における MARL によって得られる一方の協調的方策に着目し，実験結果間の比較を可能とするため，競争関係にある一方を一定の合理性を持った既定のルールにより行動させることとする．

2.6 報酬設計法

強化学習では，エージェントは学習の過程で得られる報酬に基づいて方策を獲得するため，学習エージェントの方策は，どのような条件でどれだけの報酬を与えるかによって決定づけられることとなる．それぞれが独立して意思決定を行う自律エージェントが学習の結果，自然と他のエージェントと協調してチームの目的を達成するような方策を獲得させるためには，協調することによって報酬が得られるようなインセンティブ設計が必要となる．しかしながら，複数のエージェントが同時に学習する場合，共同行動の結果に対する個々の学習エージェントの貢献度に応じた個別の報酬をフィードバックとして与えるような設計は，貢献度分配問題に帰着し [7]，汎用的な設計は容易ではない．

一般的に MARL で用いられる基本的な報酬設計法として，global reward (G) と local reward (L) が挙げられる．

global reward とは全てのエージェントに同じ報酬を与える方法である．一般に協調的な方策獲得を狙う MARL では global reward によってチームとしての望ましい状態を定義し，学習を通じて global reward を最大化する方策を得ることが望まれる．例えば，サッカーをする学習エージェントを MARL で得たい場合に，11 個の学習エージェントに対し等しく「得点したら +1，失点したら-1」といった報酬を与えるだけで，高度なチームプレイを行うような強いチームを構成する 11 個のエージェント方策が MARL によって得られることが望ましい．実際，簡単な対象問題では global reward が用いられることが多い．しかしなが

ら，global reward では対象問題の複雑性が増大するにつれ，個々のエージェントが自分自身の行動に対する適切なフィードバックが得られなくなり，学習することが困難になる．先述のサッカーの例では，得点した場合にゴールを決めたエージェント以外にも報酬が入るため，チームメイトがゴールを決めた瞬間に，その得点とは全く関係なく，他のエージェントが偶然行っていた行動が強化されるようなことが発生する．

マルコフ決定過程（MDP）において，協調エージェント $i \in N = \{1, \dots n\}$ が存在するとして，各エージェントの状態を $s_i \in S$ と仮定する．エージェント i の報酬を r_i，状態から決定される報酬関数を R とすると global reward (G) は以下のように表される．

$$r_i = R(S). \tag{2.13}$$

一方，報酬を等分しないとした場合に極端な方法の一つは，完全に個々のエージェントのふるまいだけで特定のエージェントに与える報酬を評価する方法であり，これを local reward (L) と呼ぶ．local reward は次のように表される．

$$r_i = R(s_i) \tag{2.14}$$

これにより怠惰なエージェントは生じにくくなるが，他のエージェントに協力する合理的なインセンティブがなく，貪欲なふるまいが学習される可能性が高い．先述のサッカーの例を用いれば，ゴールを決めたエージェントのみに報酬を与えるような方法が local reward に相当し，エージェントは仲間にパスを出したりせず，全員が自分でゴールしようとするような行動を学習するだろう．このように，MARL における報酬設計は，結局のところ全体として共同した結果を個々のエージェントの評価にどのように結びつけるべきかという貢献度分配問題（credit assignment problem）となる [7].

一方，Tumer らは強化学習において分権的協調の発現を設計することを COllective INtelligence(COIN) と呼び，Wonderful life utility [*1] と呼ぶ効用関数を提案した [37]．これは，あるエージェントが仮に存在しなかった場合に社会的効用がどうなっていたかを考慮するものである．この効用関数に基づいて，あるエージェント i の報酬を考えるとき，システム全体の効用と，そのエージェント i を除いた時の効用の差分を報酬とすることでエージェント i の貢献を評価する difference reward (D) と呼ぶ MARL の報酬設計を提案している [38, 39]．difference reward はシステム全体の状態 S のうちエージェント i に依存しない状態成分を S_{-i} と表す場合，式(2.15)のように表される．

$$r_i = R(S) - R(S_{-i}) \tag{2.15}$$

（なお状態ベクトルの結合を $S = S_i + S_{-i}$ のように表現する．）difference reward を用いて，連続空間でノイズの大きい環境への適用性検証 [40, 39] や，Potential-based reward

shaping (PBRS)[41] と組み合わせた研究 [42]，学習の探索効率化に着目した研究 [43] などへと発展している．

しかし，これらの方法は各エージェントの貢献度を直接評価し，各エージェントにインセンティブとして報酬を与えることに重点が置かれている．公平性の観点からあるエージェントの社会全体への貢献度を評価しようとすると，そのエージェントを除く外部性を評価することが重要なはずであり，改善の余地があると考える．

2.7 VCG メカニズム

メカニズムデザインは，ミクロ経済学とゲーム理論の一分野であり，複数の利己的なエージェントをいかにして効率よく取りまとめるかという問題を扱い，社会的余剰が最大となるような設計を行う [45]．ゲーム理論ではあるルールが与えられた際の結果を分析する．しかし，良い結果となるルールを設計するメカニズムデザインは逆ゲーム理論 (Inversed Game Theory) としても表現することができる [46]．Groves メカニズム [9] は，メカニズムデザインの一つの考え方であり，支配戦略誘引両立性 [47, 48, 49, 50] を持つ．

$a \in A$ を代替案とその集合，θ をエージェントの嗜好を表す個々のタイプとし，エージェントは代替案 a に対する価値 $v_i(\theta_i, a)$ 持つと仮定する．代替案 a の価格を p とし，エージェント i が $u_i(\theta_i, a) = v_i(\theta_i, a) - p_i$ で表される準線形効用を持つと仮定すると，決定規則 g は次の式の条件を満たす．

$$g(\theta) = \underset{a \in A}{\operatorname{argmax}} \sum_{i \in N} v_i(a, \theta_i), \tag{2.16}$$

ここで，N はエージェントの集合で $N = \{1, 2, \ldots, n\}$ を表す．決定規則 g は次の経済的効率性を満足する．

$$\sum_{i \in N} v_i(g(\theta), \theta_i) \geq \sum_{i \in N} v_i(a, \theta_i) \tag{2.17}$$

ここで，$\theta = (\theta_i, \theta_{-i})$ である．決定規則 g は全てのエージェントに対する，真のタイプ θ_i における価値の合計値を最大化する．Groves メカニズムでは，エージェントらは彼らの選択

[*1] Wonderful life utility という名前に違和感を禁じ得ないが，José M. Vidal の “Fundamentals of Multiagent Systems” [44] によれば，以下のような背景があるそうだ．

> フランク・キャプラの古い映画 “Wonderful life”（邦題「素晴らしき哉、人生！」）を見たことがある人もいるかも知れません。その映画の中で、ジョージ・ベイリーは、自分が存在しなかった場合の世界がどうなっていたかを知る機会を得ます。彼は、自分が存在しなかった場合の世界における社会の幸福度が，自分が存在した場合の世界よりも低いことを知ります．それによって，彼は彼の存在が社会の幸福度に対して良い影響を与えていると推定し，従って彼は自分の人生を終わらせないことを決定します．そのことによって，彼は社会の幸福度を上昇させたかったのだと結論付けられます．ジョージはいい奴だね．

に応じて支払い p を支払う．Groves メカニズムの支払い規則は次の式によって定義される．

$$p_i(\theta) = h_i(\theta_{-i}) - \sum_{j \neq i} v_j(g(\theta), \theta_j) \tag{2.18}$$

ここで，$h_i(\theta_{-i})$ は，エージェント i 以外の他のエージェントのタイプに基づく任意の関数である．Groves メカニズムは支配戦略誘因両立メカニズムであるが、収益等価ではない。つまり、エージェントが支払う支払額の合計は、メカニズムが支払わなければならない支払額の合計とは一致しない。これは、現実の世界で本メカニズムを利用する場合には不都合である．VCG メカニズムは Groves メカニズムの一種であり、より収益等価性の実現に近いものである。VCG メカニズムでは，式(2.18)において任意の関数とした $h_i(\theta_{-i})$ を以下の固有の関数で置き換える．

$$p_i(\theta) = \sum_{j \neq i} v_j(g(\theta_{-i}), \theta_j) - \sum_{j \neq i} v_j(g(\theta), \theta_j) \tag{2.19}$$

$g(\theta_{-i})$ はエージェント i が環境に存在しない場合の決定規則を表す．Groves メカニズムにおける支払い規則と比較して，任意の関数 $h_i(\theta_{-i})$ が $h_i(\theta_{-i}) = \sum_{j \neq i} v_j(g(\theta_{-i}), \theta_j)$ によって置き換えられている．つまり，エージェント i の支払いは，環境にエージェント i が存在する場合としない場合の他者の価値の総和の差分となる．

ここで式(2.15)と式(2.19)を比べると，後者は支払いのため正負に注意する必要があるが，とても似た構成になっていることが分かる．重要な差異は式(2.15)の 1 項目がエージェント i を含むシステム全体の状態で計算される一方，式(2.19)で対応する 2 項目はエージェント i の嗜好を表すタイプを含まない $\theta_{j \neq i}$ によって決まるエージェントの価値関数の総和で決定される点である．このためエージェント i にとって戦略的操作不可能という重要な違いが生じる．

この VCG メカニズムを MARL における報酬設計に反映する方法 [51] について，3章で議論する．また，difference reward 及び VCG メカニズムのいずれにおいても，「あるエージェントが存在しなかった場合」を仮定し，その状態を評価する必要が生じる．ある環境状態における社会的な価値や全体に対する報酬が，特定のエージェントの不在性を容易に仮定し評価可能な問題領域では両者の適用は容易であるが，実際にはその仮定が容易ではない問題領域も多い．この点について，4章で議論する．

2.8 結言

本章では，本研究の背景を述べるために，はじめにチームに関わる用語を整理したうえで，強化学習及びマルチエージェント強化学習（MARL）における課題と本研究が対象とする問題領域について整理し，対象問題の環境複雑性とその克服のための基礎的なアイデアにつ

いて述べた．続いて，MARL で必要となる報酬設計に関する関連研究について述べ，本章の最後にメカニズムデザインの関連研究として VCG メカニズムについて述べた．

第 3 章

Leader 指示と学習フレームワーク

3.1 序言

本章では，強化学習において同時に学習するエージェントが複数存在するマルチエージェント環境であることに加え，連続空間，部分観測及び競争的環境を対象に，マルチエージェント強化学習によって望ましいチームワークを実現するエージェントを得る方法を追求する． 特に，協調のための組織的枠組みを調査するのに適している追跡問題を対象に，1. Leader-Follower モデルの導入，2. カリキュラム学習という 2 点の提案を行う．Leader-Follower モデルの導入では，環境複雑性による困難性を克服するため，能力差のあるチームにおいて，Leader から Follower に対する一定の強制力を持った通信によって MARL でチームワークを向上する方法を提案する．また，競争的環境において「勝ち方」を教示するため， 学習の初期段階において一方のチームに対し競争関係にあるもう一方のチームが「あえて負ける」行動を教示するカリキュラム学習を提案する．

提案手法について実験を行い，捕食者による獲物の捕獲回数等の性能的な確認に加え，学習によって獲得された方策に基づき，エージェントの具体的な行動を観察し，従来手法と比較して，提案手法の有効性を確認する．

3.2 追跡問題の設定

3.2.1 MPE の追跡問題

本章では，エージェント間の組織的関係や通信が協調的作業に与える影響について調査するため，追跡問題 (Predator-Prey) を取り上げる．追跡問題は，協調のための組織的枠組みを調査するのに適している [52, 53] ． 2.4節で述べた MADDPG の論文 “Multi-Agent Actor-Critic for Mixed Cooperative-Competitive Environments” [1] では，手法優位性を確認するための環境として Multi-Agent Particle Environment (MPE) という複数のマルチエージェント・シミュレーション環境を用意しており，OpenAI がオープン・ソース

表3.1: MPE の追跡問題の基本設定

	Prey	Predator
エージェント数 n	1	3
サイズ	0.05	0.075
加速度倍数 (accel)	4.0	3.0
最大速度 (max_vel)	1.3	1.0

として公開している [54]．MPE には連続空間を対象とした追跡問題として，simple-tag と呼ばれる環境が含まれており，この MPE のsimple-tag 環境を用いた研究例は数多くある．例えば [55] では MADDPG で得られる協調行動を分析しているが，協調行動を促進する提案などは含まれない．Minimax 法を適用し悲観的に自身の方策を更新する手法 [56] や Attention Module と呼ぶ一元的な Critic を用いる手法 [57]，Generative adversarial imitation learning(GAIL) を適用した例 [58] などでも提案手法を評価する対象として MPE の追跡問題を用いている．しかし，いずれも学習器そのものを変更する提案であり，学習機構が複雑化することで特定の問題への性能向上と引き換えに，汎用性低下やパラメータ増加などの懸念がある．本論文では，学習機構はシンプルに留め，エージェントが入力する観測情報 o とエージェントが出力する行動 $\boldsymbol{a}$ を変更することでチームの協調方策を向上させることを試みる．提案手法を説明する前に，まず前提となる MPE における追跡問題がどのように定義されているかを以下にまとめる．

MPE の追跡問題では，各エージェント i の観測情報 o_i は式(3.1)，行動 $\boldsymbol{a_i}$ 及び行動に基づいて決まるエージェントに印加される力 f_i^a はそれぞれ式(3.4)及び式(3.5)のように表される．

$$o_i = \{p_i, v_i, A_i, B_i\} \tag{3.1}$$
$$A_i = \{p_j - p_i \mid (\forall j \in X) \wedge (j \neq i)\} \tag{3.2}$$
$$B_i = \{v_j \mid (\forall j \in X) \wedge (j \neq i)\} \tag{3.3}$$
$$\boldsymbol{a_i} = [a_i^1, a_i^2, \ldots, a_i^5] \tag{3.4}$$
$$f_i^a = \mathrm{accel}_i \times \left(\begin{bmatrix} a_i^2 \\ a_i^4 \end{bmatrix} - \begin{bmatrix} a_i^3 \\ a_i^5 \end{bmatrix} \right) \tag{3.5}$$

ここで，Prey と Predator からなるエージェントの集合を X，エージェント i の位置を $p_i = \begin{bmatrix} x_i \\ y_i \end{bmatrix}$，速度を $v_i = \begin{bmatrix} \dot{x_i} \\ \dot{y_i} \end{bmatrix}$，エージェント i からみた他のエージェントの相対位置の集合を式(3.2)の通りA_i，i 以外のエージェントの速度の集合を式(3.3)の通りB_i とする．

また，各エージェントの諸元は表3.1に示す通りに設定される．式(3.5)のように f_i^a の計算には，$\boldsymbol{a_i}$ のうち要素 a_i^1 は使用されない．MADDPG が出力する行動空間，ここでは $\boldsymbol{a_i}$ の各要素は $[0, 1]$ である一方，$\sum_{k=1}^{5} a_i^k = 1$ となる実装であるため，冗長変数として用意されて

いるものと思われる．なお，当該時間ステップにエージェントが重なっている場合，その重なりの距離に応じて互いに反発する力が印加される．また，エージェントの速度には慣性を持たせるためにダンピング係数 d が設定されており，$v' = v(1-d), d = 0.25$ となる．また，最終的に $v' > \text{max\_vel}$ となる場合，最大速度 max_velで制約され $|v'| = \text{max\_vel}$ となる．表3.1の通り，Prey が加速度倍数，最大速度ともに Predator より優れるため，Predator は単に Prey を追いかけるだけでは捕捉することができない．

ある時間ステップに Prey と Predator が重なっている（Predator が Prey を捕捉している）場合，Prey は重なっている Predator の数 $\times(-10)$ の報酬を得，Predator も同様に Prey に重なっている Predator の数 $\times(+10)$ の報酬を得る．

また，Prey 及び Predator エージェントを領域内に留めるため，Prey にのみ $|x, y| < 0.9$ では 0，$0.9 \leq |x, y| < 1$ では $-10(|x, y| - 0.9)$，$|x, y| \geq 1$ では $-\min[\mathrm{e}^{2|x,y|-2}, 10]$ となる r^{penalty} が加算される．Prey は領域外で負の報酬を得ることにより，学習が進むにつれ x, y の両座標について $[-1, 1]$ の領域内に留まるように行動し，Prey を追う Predator も領域内で行動するようになる．

3.2.2 部分観測情報環境化

MPE の追跡問題環境は式(3.1)〜式(3.3)の通り各エージェントが常に他のエージェントの情報を取得できることを前提としている．本論文では，表3.1の Predator エージェントの数を 4 に変更し，追加する 1 つを Leader，残りを Follower と位置付ける．その上で，リーダーの存在を念頭に，Predator のエージェントに能力差を設定する．すなわち，Leader は広範囲の情報を取得できる一方，Follower は限られた範囲しか環境に関する情報を取得できないようにする．Leader は常に環境全体を観測可能なように設定し，式(3.6)及び式(3.7)の通り Follower は自身を中心に 0.2 の範囲に他のエージェントの中心位置が存在する場合に限り，他のエージェントの情報を観測可能とする．

$$
\begin{aligned}
i \in\ & \{\text{Follower}\} \\
A'_i =& \{p_j - p_i \mid (\forall j \in X) \wedge (j \neq i) \wedge P(i,j)\} \\
& \cup \{[\begin{smallmatrix}0\\0\end{smallmatrix}] \mid (\forall j \in X) \wedge (j \neq i) \wedge (\neg P(i,j))\} && (3.6) \\
B'_i =& \{v_j \mid (\forall j \in X) \wedge (j \neq i) \wedge (P(i,j))\} \\
& \cup \{[\begin{smallmatrix}0\\0\end{smallmatrix}] \mid (\forall j \in X) \wedge (j \neq i) \wedge (\neg P(i,j))\} && (3.7) \\
k \in\ & \{\text{Leader}, \text{Prey}\} \\
A'_k =& A_k,\ B'_k = B_k && (3.8) \\
o_l =& [p_l, v_l, A'_l, B'_l],\ (l \in X) && (3.9)
\end{aligned}
$$

ここで，$P(i, j)$ を Follower i とエージェント j の中心間距離が 0.2 より小さい（観測可能）な場合に真，それ以外で偽となる関数とする．観測範囲に制約がある Follower を擁する

Predator がチームとしての効用を高めるためには，エージェント間で観測能力の差を補完するように協調することが必要な問題設定となる．

3.2.3 Prey エージェント

Predator の協調的方策の獲得に着目し結果を比較可能とするため，Prey エージェントを以下のように既定のルールにより行動を決定するようにした．

図3.1に追跡問題の実行例と Prey の目標候補点について示す．Prey は図3.1に示す 4 つの点から 1 点を目標点として，敵対する全ての Predator の位置を元に決定する．ステップを 1 つ進める際の処理ごとに，4 つの目標候補点を順に走査し，候補点と全ての Predator の距離の総和を計算する．その後，4 つのうち総和が最も大きい値，すなわち最も Predator 群から遠い点を選択し，5 ステップの間，目標点に向かって目標点との距離に応じた力を印加する．Prey は一時的に Predator に捕捉されることがあったとしても，定期的に Predator 群から距離を保とうと目標点を更新するため，継続的に捕捉し続けることが困難な設定となる．

3.2.4 報酬

Y を Predator の集合，$D(i)$ をある時間ステップに Prey と Predator i が重なっている（Predator i が Prey を捕捉している）場合に 1，それ以外の場合に 0 となる関数とすると，MPE の追跡問題では Predator i の報酬は式(3.10)のように定義されている．式(3.10)により MPE の追跡問題における Predator 側の目的は「できるだけ多くの Predator ができるだけ多くのタイムステップで Prey に重なりを持つ（捕捉する）」と捉えることができる．この報酬設計は，どの Predator が Prey を捕捉しても全ての Predator に等しく報酬が与えられる，チームとしての目標を直接的に報酬に設定するもので， Global reward と呼ばれる．

一方，古典的な追跡問題 [59] では，4 つの Predator が 1 つの Prey を 2 次元グリッド空間において同時手番で上下左右に移動または静止し，四方から Prey を取り囲むことを目的としている．本研究では，古典的な追跡問題の目的を踏まえ，式(3.11)のように 4 つの Predator が同一タイムステップに Prey を捕捉した場合（以下，全同時捕捉と呼ぶ）には r^{bonus} を加算することとし，その状態がどの程度生起するかを観察する．また，Follower は3.2.2節に示したように非常に限定された観測範囲しか持たず，領域外に出てしまっては他のエージェントを観測する機会がほとんどないため，本実験ではいわば領域に関する事前知識として Predator 側にも3.2.1節で示した r^{penalty} を与えることとする．以上により本実験では，式(3.12)の報酬を用いる．

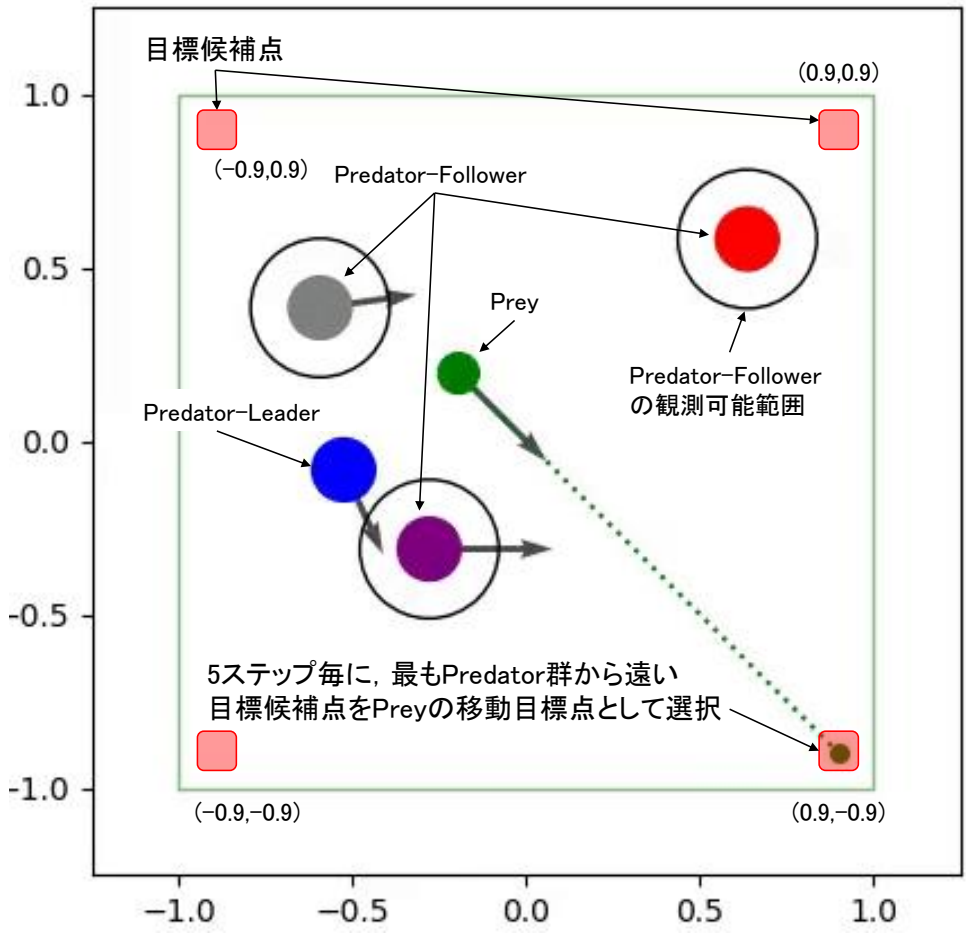


図3.1: 追跡問題の実行例と Prey の目標候補点

$$r_i = \sum_{j \in Y} D(j) \times 10 \quad :\text{MPE} \tag{3.10}$$

$$r^{\text{bonus}} = \begin{cases} 50 & \text{if } D(\forall j \in Y) = 1 \\ 0 & \text{else} \end{cases} \tag{3.11}$$

$$r'_i = \sum_{j \in Y} 10D(j) + r^{\text{bonus}} + r_i^{\text{penalty}} \tag{3.12}$$

3.3 提案手法

3.3.1 Leader 指示と Leader 強制力

常に環境全体を観測可能な Leader i が Follower j に対し指示を行えるよう，式(3.13)の通り，Leader は式(3.4)の $\boldsymbol{a}$ に加え 7 つの要素を追加出力する．式(3.14)及び式(3.15)の通り 2 変数の行列 f_j^l を決定し，式(3.16)のように Follower に印加する力に加算することで，

Leader 指示に一定の強制力（Leader 強制力）を付与する．Leader 強制力の大きさによる影響を確認するため，強制力の大きさを決定する係数 α を設ける．さらに，式(3.17)の通り f_j^l を Follower の観測情報 o' に加えることで，Leader から Follower に対する一方向通信（Leader 指示）となる．式(3.15)に示した Leader 指示は，それぞれ図3.1の座標系で $c = 6$: 何もしない，$c = 7$: 左，$c = 8$: 右，$c = 9$: 下，$c = 10$: 上，$c = 11$:Leader の方向，$c = 12$:Leader とは逆方向，の 7 つの指示に相当する．7 通りという非常に少ない情報量の指示としたのは，前述のサッカーの例のように短時間で非常に少ない情報量しか伝達できない事例を想定したものである．なお，$c = 12$（Leader から離れる力を印加）の場合は，対象の Follower が領域の端に存在する場合には領域内に留まらせるために力を印加しないようにした．

提案手法は DDPG と組み合わせるものとし，環境の 1 ステップ更新に関わる処理の流れについて Algorithm 1 に，あるタイムステップ t における Leader L から Follower F_1 に対する Leader 指示及び Leader 強制力 $f_{F_1}^l(t)$ の与え方を図3.2に示す．

$$\boldsymbol{a_i'} = [[a_i^1, a_i^2, \ldots, a_i^5], [a_i^6, a_i^7, \ldots, a_i^{12}]] \tag{3.13}$$

$$c = \underset{k \in \{6,7,\ldots,12\}}{\operatorname{argmax}} [a_i^k] \tag{3.14}$$

$$f_j^l = \begin{cases} \begin{bmatrix} 0 \\ 0 \end{bmatrix} & \text{if } c = 6 \\ \begin{bmatrix} -1 \\ 0 \end{bmatrix} & \text{if } c = 7 \\ \begin{bmatrix} 1 \\ 0 \end{bmatrix} & \text{if } c = 8 \\ \begin{bmatrix} 0 \\ -1 \end{bmatrix} & \text{if } c = 9 \\ \begin{bmatrix} 0 \\ 1 \end{bmatrix} & \text{if } c = 10 \\ -(p_j - p_i) & \text{if } c = 11 \\ (p_j - p_i) & \text{if } c = 12 \wedge ((|x_j| \leq 0.9) \wedge (|y_j| \leq 0.9)) \\ \begin{bmatrix} 0 \\ 0 \end{bmatrix} & \text{if } c = 12 \wedge ((|x_j| > 0.9) \vee (|y_j| > 0.9)) \end{cases} \tag{3.15}$$

$$f_j'^a = f_j^a + \alpha f_j^l \tag{3.16}$$

$$o_j' = \{o_j, f_j^l\} \tag{3.17}$$

Algorithm 1: Calculate environment step

Leader and Followers select action w.r.t. the current policy and exploration*1
Prey select action as described at section 3.2.3
Calculate action force f^a according to equation (3.5)*1
for j $\in$ {Follower} **do**
    calculate f_j^l according to equation (3.14) and (3.15)
end for
Set $f^l = [\begin{smallmatrix}0\\0\end{smallmatrix}]$ for Prey and Leader
Calculate force f_i^e when they are in collision*1
Integrate force $f_i = f_i^a + f_i^l + f_i^e$ *1
Calculate $v_i' = v_i(1 - \text{damping}) + \frac{f_i}{\text{mass}} \times \mathrm{d}t$ (damping= 0.25, mass= 1, $\mathrm{d}t = 0.1$)*1
if $|v_i'|$ >max_speed$_i$*1 **then**
    $v_i' := \frac{v_i'}{|v_i'|} \times$ max_speed$_i$ *1
end if
Calculate new position : $p_i' = p_i + v_i' \times \mathrm{d}t$ *1
Calculate reward r according to equation (3.12)
Set new state according to equation (3.9) and (3.17)

*1 Same as originally designed in MADDPG

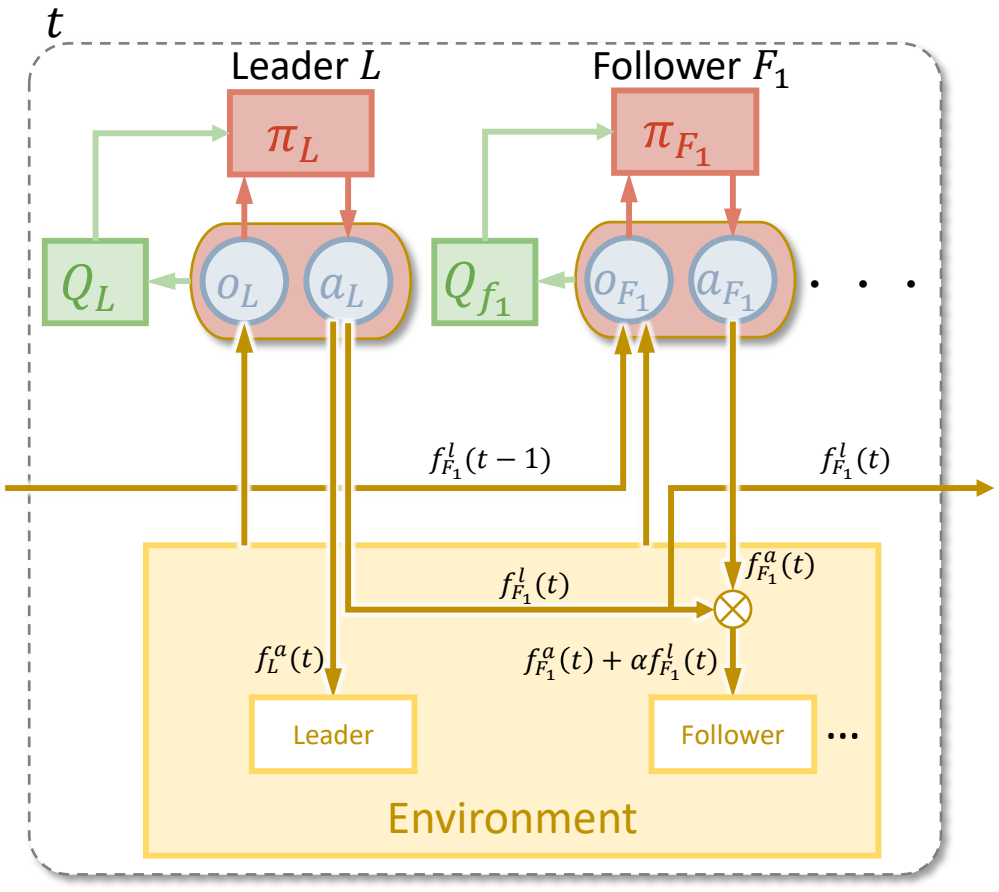


図3.2: タイムステップ t における Leader$L$ から FollowerF_1 に対する Leader 指示及び Leader 強制力 $f_{F_1}^l(t)$ の与え方

3.3.2 カリキュラム学習

カリキュラム学習 [60] とは，学習の難易度を徐々に上げていく学習手法である．最初から目的とするタスクに向けた訓練データを与えるのではなく，相対的に容易に達成可能なタスクから学習を行うことで良好な解への収束が早くなることが実験的に示されている．例えば Bengio らは，グレースケール画像に矩形，楕円，及び三角形のうちどれが描かれているのかを分類する問題を取り上げている．実験では，最初はそれぞれの図形の形状，位置，サイズなどの要素のバリエーションが少ないデータセットに対して学習させ，次に各要素のバリエーションを増やしたデータセットに対して学習させるというカリキュラム学習を行い，カリキュラム学習を行った場合の方が，最初から各要素のバリエーションの多いデータセットに対して学習させるよりも識別エラー率が低くなるという結果を示した．ただし，どのようなカリキュラムを構成するかは設計者に依存しており，対象とする問題領域に依存しない汎用的なカリキュラム構成の原則の解明は今後の課題としている．

そこで，追跡問題に対する強化学習においてどのようなカリキュラム構成とするのが効果的か検討する．学習の初期にはエージェントはほぼランダムな行動を行うため，Predator と Prey が接触することは稀である．特に今回のように Prey がルールによって行動し学習の初期から積極的に Predator を回避する場合，偶然性によって Predator と Prey が接触し，Predator のエージェントが Prey を捕捉することによって報酬を得ることができるということを知る機会は非常に少ないといえる．そこで，学習の初期段階では Prey 自ら Predator に順番に接触していく，言い換えれば Prey が「あえて負ける」というカリキュラムを適用して，Predator が報酬を得る機会を与えることによって，学習を通じて積極的に Prey の捕捉を試みる方策を得る助けになると考えた．式(3.18)〜式(3.21)のように Prey の行動を決定することで，「あえて負ける」というカリキュラムの間，Prey は各 Predator に対して順に向かっていく加速度が与えられる．

$$\boldsymbol{a}^2_{\text{prey}} = \begin{cases} \min\left((x_i - x_{\text{prey}}) \times 3.5, 1.3\right) & \text{if } x_i - x_{\text{prey}} \geq 0 \\ 0 & \text{otherwise} \end{cases} \tag{3.18}$$

$$\boldsymbol{a}^3_{\text{prey}} = \begin{cases} \min(|x_i - x_{\text{prey}}| \times 3.5, 1.3) & \text{if } x_i - x_{\text{prey}} < 0 \\ 0 & \text{otherwise} \end{cases} \tag{3.19}$$

$$\boldsymbol{a}^4_{\text{prey}} = \begin{cases} \min\left((y_i - y_{\text{prey}}) \times 3.5, 1.3\right) & \text{if } y_i - y_{\text{prey}} \geq 0 \\ 0 & \text{otherwise} \end{cases} \tag{3.20}$$

$$\boldsymbol{a}^5_{\text{prey}} = \begin{cases} \min(|y_i - y_{\text{prey}}| \times 3.5, 1.3) & \text{if } y_i - y_{\text{prey}} < 0 \\ 0 & \text{otherwise} \end{cases} \tag{3.21}$$

ここで，式(3.18)〜式(3.21)における i は各 Predator を指定し，15 ステップごとに Prey が向かう対象とする Predator を切り替える．カリキュラム切り替えは，総エピソード数に

表3.2: 学習パラメータの設定

trainer	DDPG	learning rate	1.0e-2
gamma	0.95	batch size	1024
hidden units	64	num layers	2

対し十分小さく，かつ全 Predator が十分な回数 Prey との接触を経験すると考えた 5,000 エピソードとした．

3.4 実験

3.4.1 実験ケース

学習器には基本的な深層強化学習手法として連続量の行動が出力可能な Deep Deterministic Policy Gradient (DDPG) [20] と MADDPG を用い，全ての実験は Lowe らによる実験 [1] の通りの学習アルゴリズム実装と表3.2のパラメータを用いる．

提案手法の有効性を確認するため，1 エピソード 50 ステップとして 100,000 エピソードの学習を表3.3に示す 7 ケースで各 10 回ずつ実施する．ケース (1) を提案手法として式(3.16)において $\alpha = 1.0$ としたケースとする．次に Leader から Follower に対し情報を与えることの影響を確認するため，ケース (2) を Follower に対し Leader 強制力 f_j^l はケース (1) と同様 $\alpha = 1.0$ として付与するが，Follower の観測情報 o' には与えない，すなわち式(3.17)を式(3.22)のように変更したケースとする．

$$o'_j = \{o_j, \begin{bmatrix} 0 \\ 0 \end{bmatrix}\} \tag{3.22}$$

また，ケース (3), (4) を提案手法のうち Leader 強制力の大きさによる影響を確認するため，それぞれ α を 0.5 0 としたケースとする．ケース (5) を提案手法のうちカリキュラム学習を行わない場合とし，各結果を比較することで提案手法の各要素の有効性を確認する．さらに，提案手法は DDPG と組み合わせることとしているが，学習アルゴリズムによる違いを確認するため，ケース (6) としてあえて MADDPG と提案手法を組み合わせた場合，ケース (7) として提案手法を適用しない MADDPG による学習を行い，提案手法と結果を比較する．

3.4.2 評価方法

各実験ケースの 10 回の比較を行うため，母数が少なくても検定が可能であるノンパラメトリック検定を用いる．本研究では提案手法のうち Leader から Follower に対する通信有無（Leader 指示），Leader 強制力の大きさ，カリキュラム学習の有無，それぞれの要素によ

表3.3: 実験ケース別の設定

<table>
<tr><th rowspan="2">番号</th><th colspan="2">Leader-Follower モデル</th><th rowspan="2">カリキュラム</th><th rowspan="2">アルゴリズム</th></tr>
<tr><th>Leader 指示</th><th>Leader 強制力</th></tr>
<tr><td>(1)</td><td>あり</td><td>1.0</td><td rowspan="4">あり</td><td rowspan="5">DDPG</td></tr>
<tr><td>(2)</td><td>なし</td><td>1.0</td></tr>
<tr><td>(3)</td><td rowspan="3">あり</td><td>0.5</td></tr>
<tr><td>(4)</td><td>0</td></tr>
<tr><td>(5)</td><td>1.0</td><td>なし</td></tr>
<tr><td>(6)</td><td></td><td>1.0</td><td>あり</td><td rowspan="2">MADDPG</td></tr>
<tr><td>(7)</td><td colspan="3">なし</td></tr>
</table>

り捕捉回数に影響を与えているかを調査する．ケース (1) 提案手法，ケース (2) Leader 指示なし，ケース (3) Leader 強制力半分，ケース (4)Leader 強制力なし，及びケース (5) カリキュラムなしの 5 群に対応した多重比較の検定である Kruskal-Wallis 検定を用いる．また，提案手法であるケース (1) とその他のケースの比較では 2 組の標本の有意差検定に対応した Wilcoxon の順位和検定を用いる．

3.4.3 学習の推移

図3.3〜図3.9 に各ケースの学習の推移を示す．各ケースのグラフはそれぞれ，上段が学習時の 1,000 エピソード毎の各エージェントの累積報酬の平均値と標準偏差を，中段が学習中の evaluation による Predator 毎の Prey 捕捉平均値の積み上げ，下段が同じく evaluation 中に生起した全同時捕捉回数の平均値を示す．

各グラフ上段から，DDPG を用いたケース (1) 図3.3〜ケース (5) 図3.7では学習が収束していることが分かる．ケース (1) 図3.3〜ケース (5) 図3.7を比較すると，ケース (1) 図3.3 及びケース (2) 図3.4ではグラフ中段の Leader と 3 つの Follower の捕捉回数のばらつきが小さく，一方でケース (4) 図3.6 ではほとんど Leader のみが捕捉するような方策が得られていることが示唆される．Predator の捕捉傾向に偏りがあるとグラフ下段に示す全同時捕捉は生起し難く，Leader 強制力のあるケース (1) 図3.3，ケース (2) 図3.4，ケース (3) 図3.5及びケース (5) 図3.7でのみ生起している．さらに，Leader 強制力の大きさが異なるケース (1) 図3.3，ケース (3) 図3.5及びケース (4) 図3.6を比較すると，Leader 強制力が小さくなるにつれ Follower の捕捉回数が減少している．このことから，Leader 強制力が報酬の増加に寄与していることが分かる．一方，Leader 指示の有無が異なるケース (1) 図3.3とケース (2) 図3.4の比較では，報酬の獲得傾向に大きな違いは見られない．ケース (1) 図3.3とカリキュラム学習を行わなかったケース (5) 図3.7を比較すると，ケース (5) 図3.7では Follower の捕捉回数に偏りが大きく，カリキュラム学習が Predator チームの報酬増加に効果があること

が分かる．

MADDPG を用いたケース (6) 及び (7) では報酬が単調増加傾向になっておらず，学習が安定していない．ケース (6) と (7) の Prey の捕捉回数（グラフ中段）を比較すると，Leader 指示，Leader 強制力及びカリキュラム学習を適用しない (7) ではほとんど Leader のみが捕捉しており，適用したケース (6) ではわずかに Follower も捕捉に参加していることがうかがわれるという違いがある．そしてケース (6) 及び (7) ともに，ケース (1) との比較では明らかに Prey の捕捉回数に差がある．以上から，MADDPG は部分観測情報に対応できないことが本実験でも確認された．MADDPG では各エージェントの Q を推定する Critic に対して他のエージェントの観測情報と行動が入力されるが，部分観測情報環境では無効な観測情報と行動のセットを得る機会が多くなり，むしろ学習に対してノイズになるものと考えられる．この点については，Wang らが R-MADDPG の提案 [30] においても指摘しており，Wang らはリカレント・ニューラルネットを用いて観測情報の時系列的な推移を考慮することで影響を緩和できることを主張している．

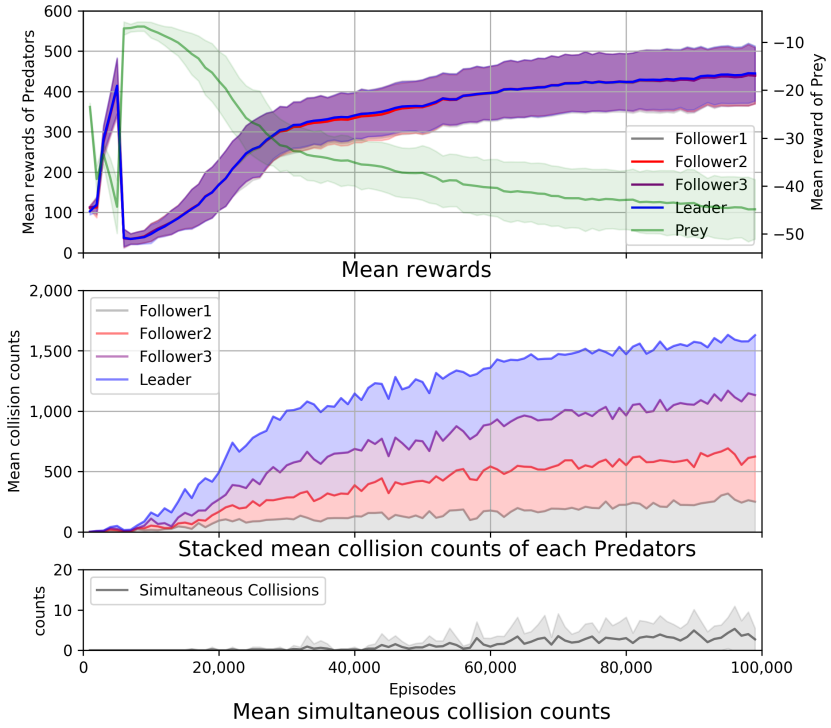


図3.3: ケース (1) 提案手法
学習時の評価値と報酬値の推移

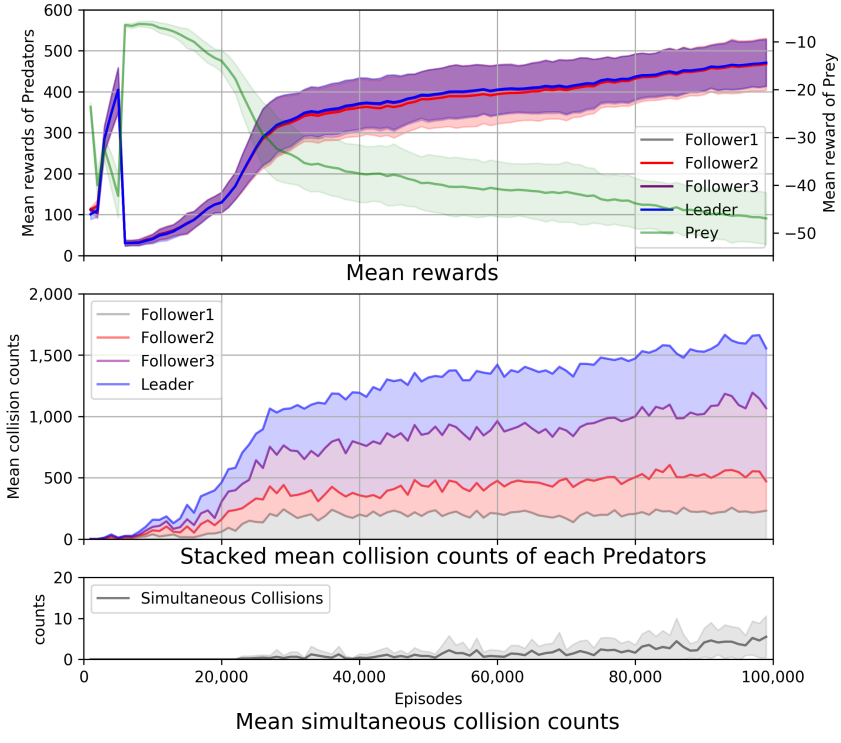


図3.4: ケース (2) Leader 指示なし
学習時の評価値と報酬値の推移

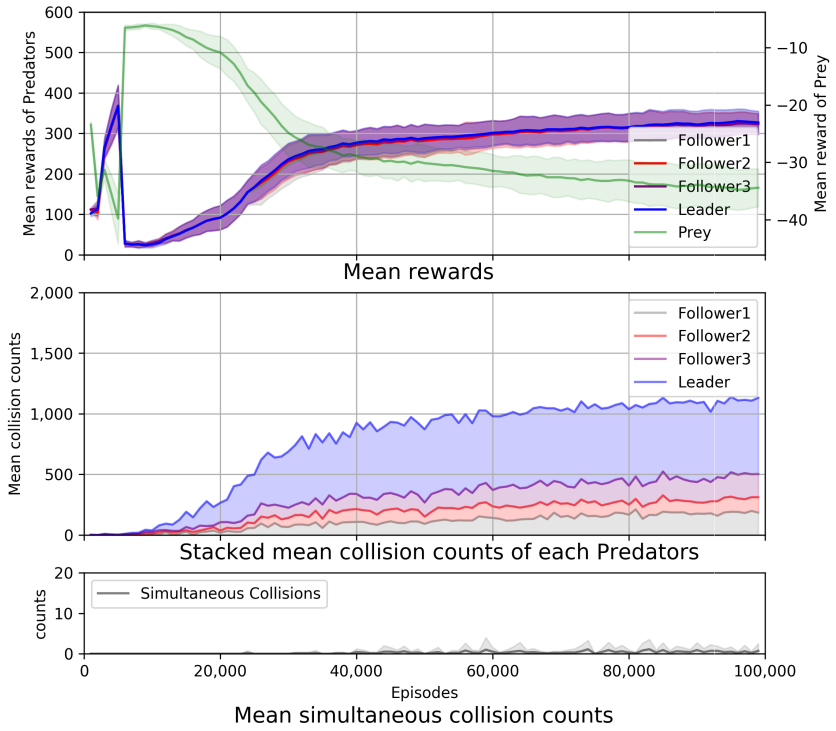


図3.5: ケース (3) Leader 強制力半分
学習時の評価値と報酬値の推移

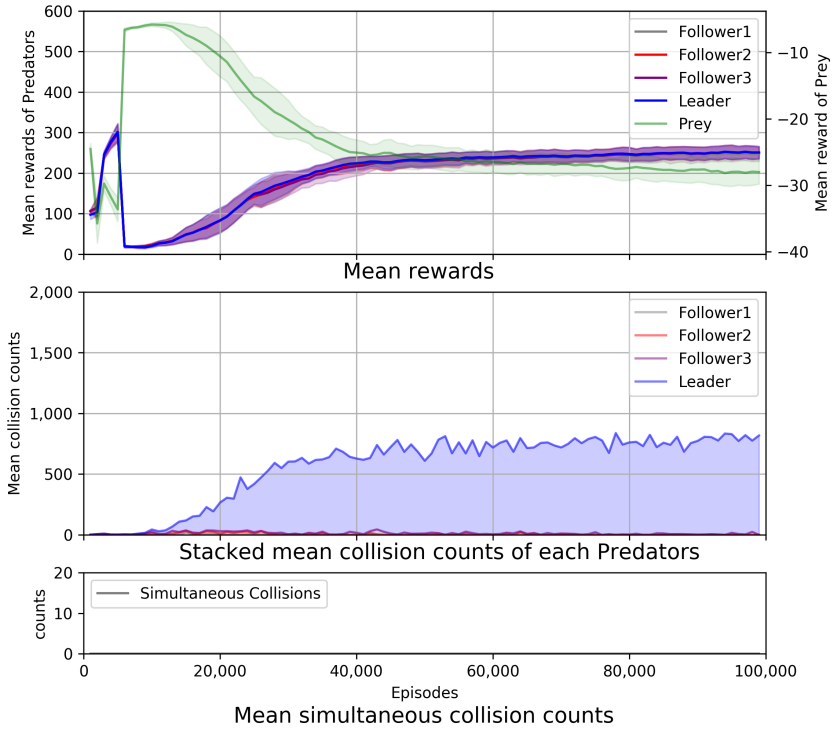


図3.6: ケース (4) leader 強制力なし
学習時の評価値と報酬値の推移

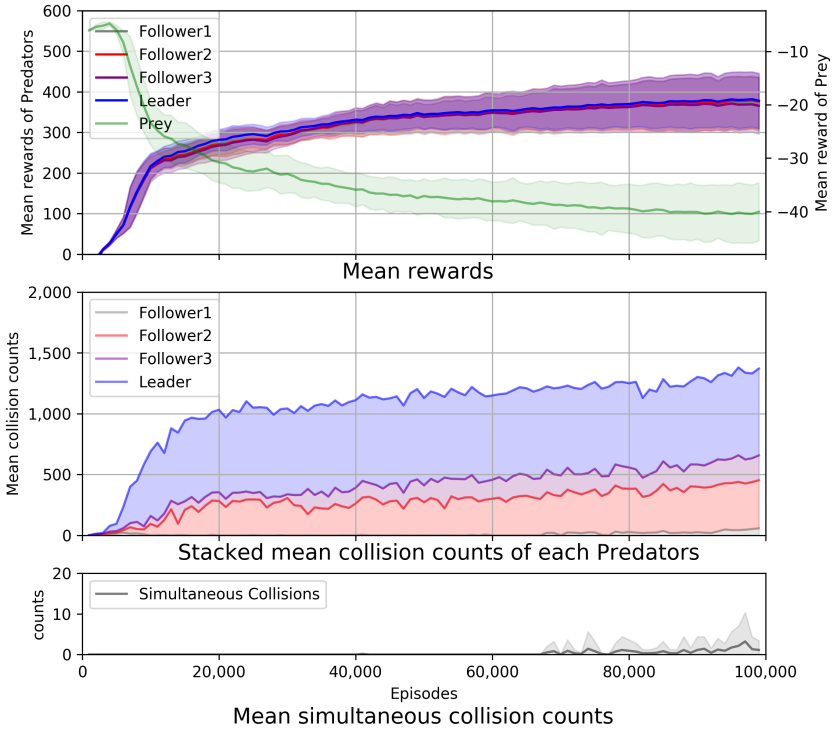


図3.7: ケース (5) カリキュラムなし
学習時の評価値と報酬値の推移

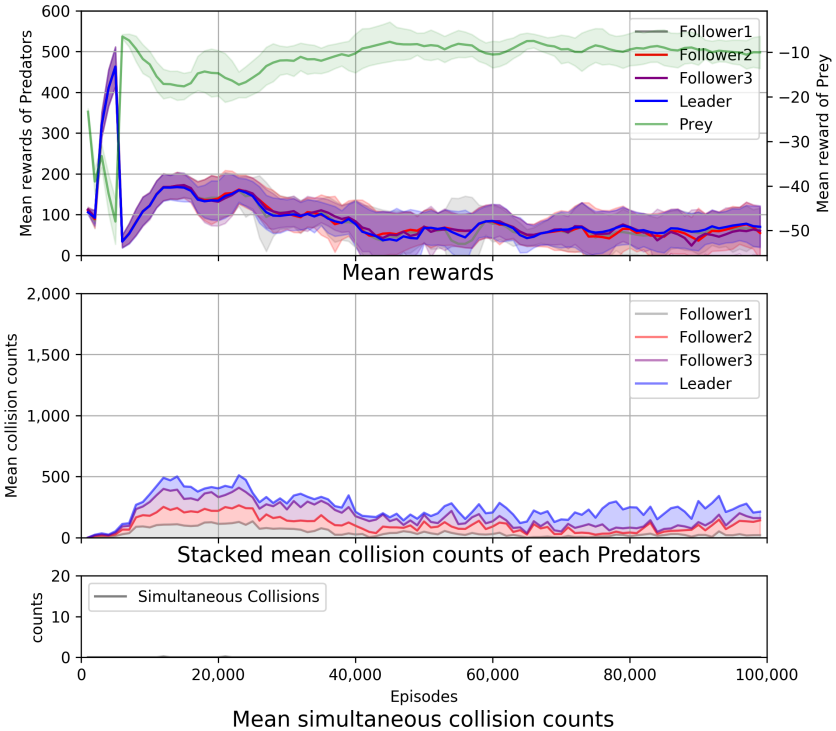


図3.8: ケース (6) 提案手法と MADDPG
学習時の評価値と報酬値の推移

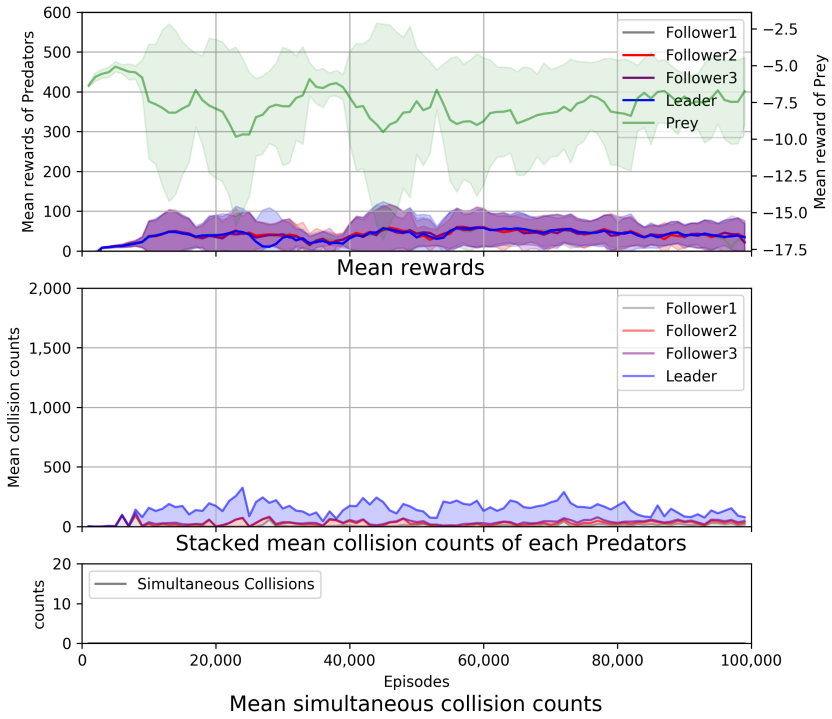


図3.9: ケース (7) MADDPG
学習時の評価値と報酬値の推移

一般に強化学習の学習の良否は報酬値が単調増加する様子で示すことが多いが，競争環境では互いに戦略を変化させるため単調増加とはならない．このような学習環境では，最高の評価値となる学習済みモデルを保存することが有効であることが分かる．なお，本実験では3.2.3節に示したように Prey エージェントはルールで行動を決定しており学習はしていないが，図3.3 〜図3.9 には Prey が捕捉される毎に −10 とした値を報酬値として参考に記載した．

3.5 結果の分析

3.5.1 ベンチマーク

提案手法の各要素の有効性を比較するため，各ケースの学習によって最終的に得られた学習済みモデルによる Prey の捕捉回数を比較した結果を図3.11に，同提案手法と MADDPG を用いたケースとの比較を図3.12に示す．各図は横軸が表3.3に示した各実験ケース，縦左軸が平均捕捉回数，縦右軸が平均全同時捕捉回数である．なお，各学習試行において 1,000

エピソード毎にモデルを保存し，得られた全てのモデルに対し 1,000 ステップ分の追跡問題を実行して最大の Prey の捕捉回数となったモデルを，各学習試行における学習済みモデルとして採用した．なお全てのケースで公正を期すため，ベンチマーク時にはエージェントの初期位置は図3.10に示す位置に固定した．

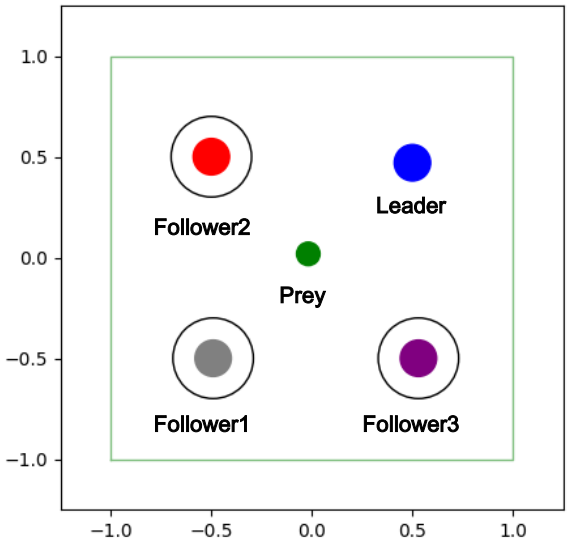


図3.10: evaluation 時のエージェント初期位置

同一ステップに複数の Predator が Prey を捕捉した場合は捕捉した Predator の数だけ捕捉回数としてカウントするため，捕捉回数が 1,000 を超えるケースでは平均して常に 1 つ以上の Predator が Prey を捕捉していることになる．図3.11では全てのケースで 1,000 を超える Prey の捕捉回数となっており，いずれも高い性能といえる．

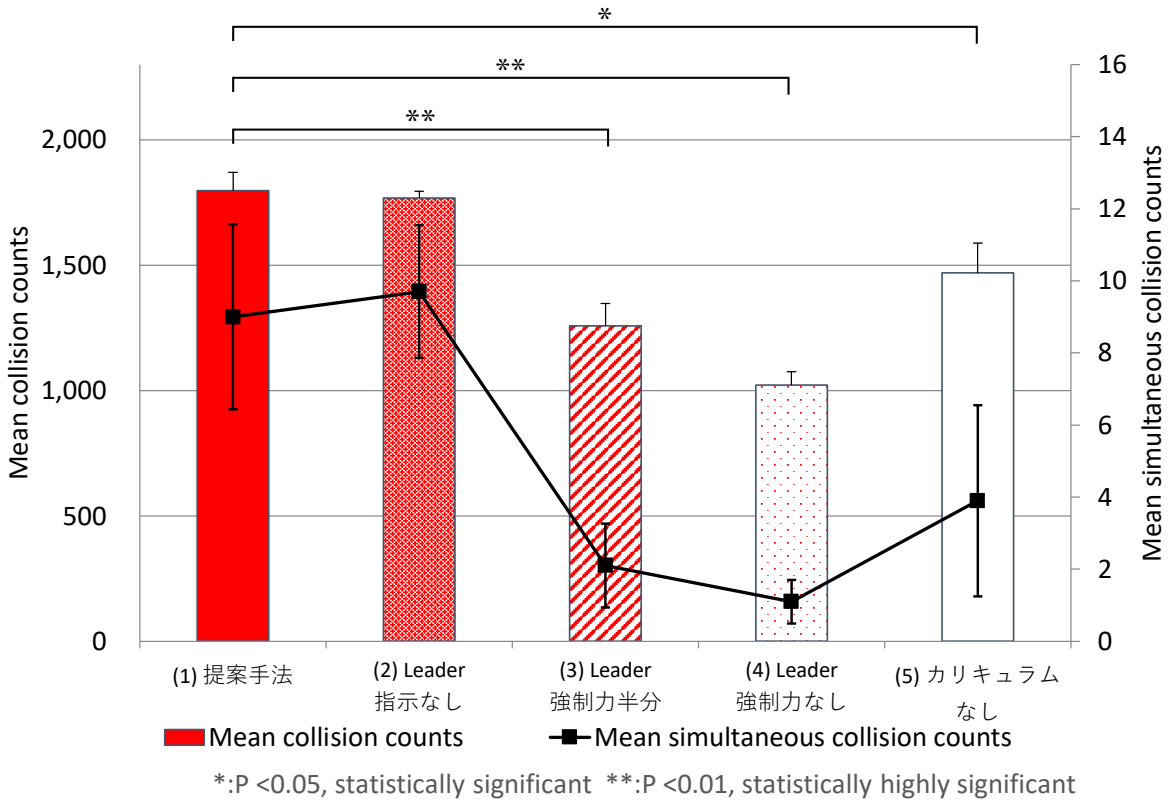


図3.11: Prey の捕捉回数（DDPG を用いた提案手法の各要素の比較）

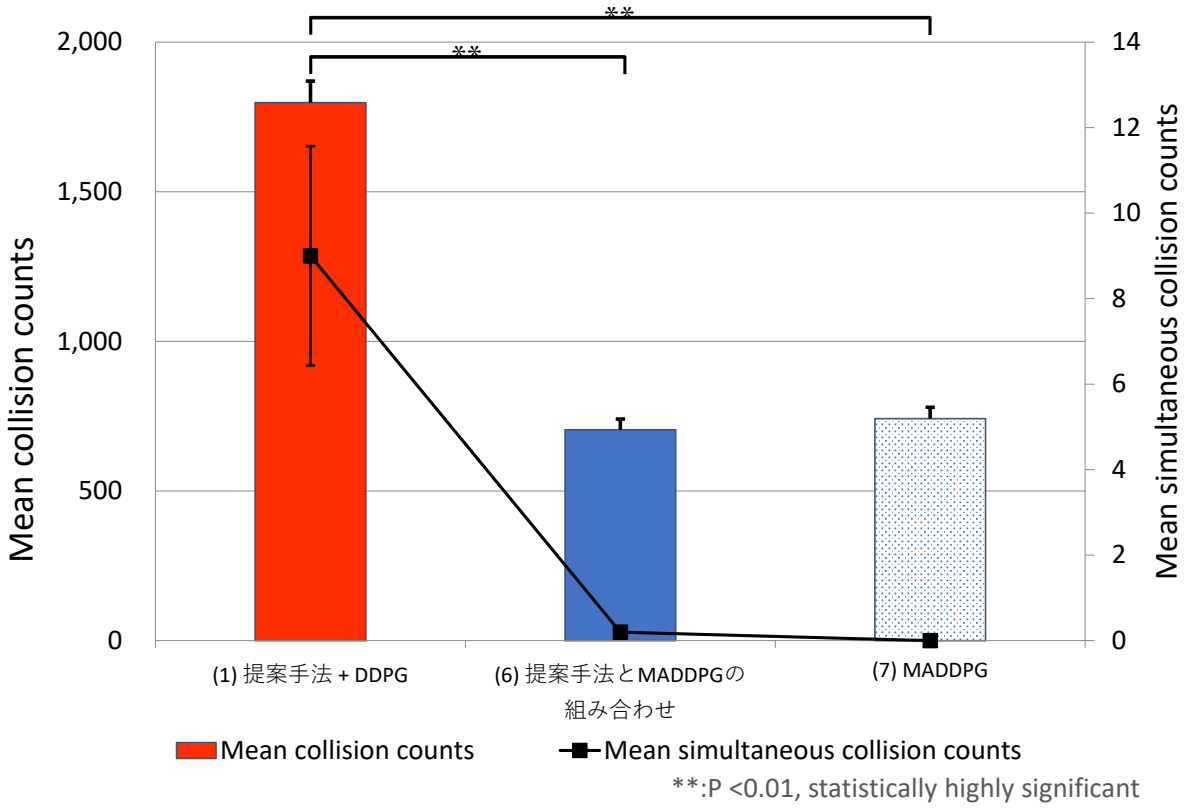


図3.12: Prey の捕捉回数（DDPG を用いた提案手法と MADDPG の比較）

Prey の捕捉回数での比較では，5 つの結果は Kruskal-Wallis 検定で $P < 0.0001$ となり有意差が認められた．また，各手法について Wilcoxon の順位和検定を行いケース (1)，(2) の比較で $P = 0.21$ となり有意水準 5% で有意差はみられなかったが，ケース (1) と (3) の比較で $P = 0.0010$，ケース (1) と (4) の比較で $P = 0.00025$，ケース (1) と (5) の比較で $P = 0.031$ となり，これらの間では有意差が確認された．Leader 強制力の大きさが異なるケース (1)，(3)，(4) の比較では，Leader 強制力が小さくなるにつれ捕捉回数が減少している．この結果を以下のようにまとめる．

- ケース (1) と (2) の比較から，Prey の捕捉回数に Leader 指示の有無による有意差は確認されない
- ケース (1), (3), (4) の比較から，Leader 強制力が大きいほど捕捉回数が有意に増加している
- ケース (1) と (5) の比較から，カリキュラムにより捕捉回数が有意に増加している

以上から，Leader 強制力の効果は大きいということが分かる．

また，図3.12の比較においても Wilcoxon の順位和検定でケース (1) と (6) 及びケース (1) と (7) の比較ともに $P = 0.00018$ であり有意差が確認できる．一方，ケース (6) と (7) の間ではほとんど差がなく，Leader 指示，Leader 強制力及びカリキュラム学習の適用は MADDPG との組み合わせではあまり効果がないことが分かる．提案手法と MADDPG の結果には大きな差が生じているが，捕捉回数のベンチマーク値だけでは差が生じた理由までは判断し難いため，次節において各エージェントがどのように行動したかを具体的に観察することで Predator のチームとしての協調方策の傾向について分析を試みる．

3.5.2 移動軌跡によるエージェント毎の行動分析

ここでは，3.5.1節において DDPG を用いた提案手法と MADDPG のベンチマーク結果の比較において大きな差が確認されたことから，各エージェントの連続的な一連の行動傾向を可視的に分析することを試みる（動画：[61]）．図3.13 にケース (1) と (7) それぞれで得られたモデルによる 1,000 ステップの移動軌跡例を示す．各図は上段が環境座標上の移動軌跡で，下段は Prey の位置を中心とした場合の Prey に対する相対位置の軌跡を示している．図3.13(c) によって，ケース (1) では全ての Predator が常時 Prey の付近に存在していることが分かる．一方 (b) 及び (d) のケース (7) では Leader のみが捕捉している．その他の Predator は積極的には Prey を捕捉しに行かず，Predator 群の位置関係によって移動する Prey の移動目標点の選択肢を限定させる役割を担っている．いわば Predator チーム内に捕捉役と誘導役という明確な役割分担が生起しているといえる． 役割分担を行うチーム方策に陥ることは本実験環境における局所解の一例であり，MADDPG よりも，Leader 指示と Leader 強制力にシンプルな学習アルゴリズムである DDPG を組み合わせた提案手法の方が効果的であることが確認できた．

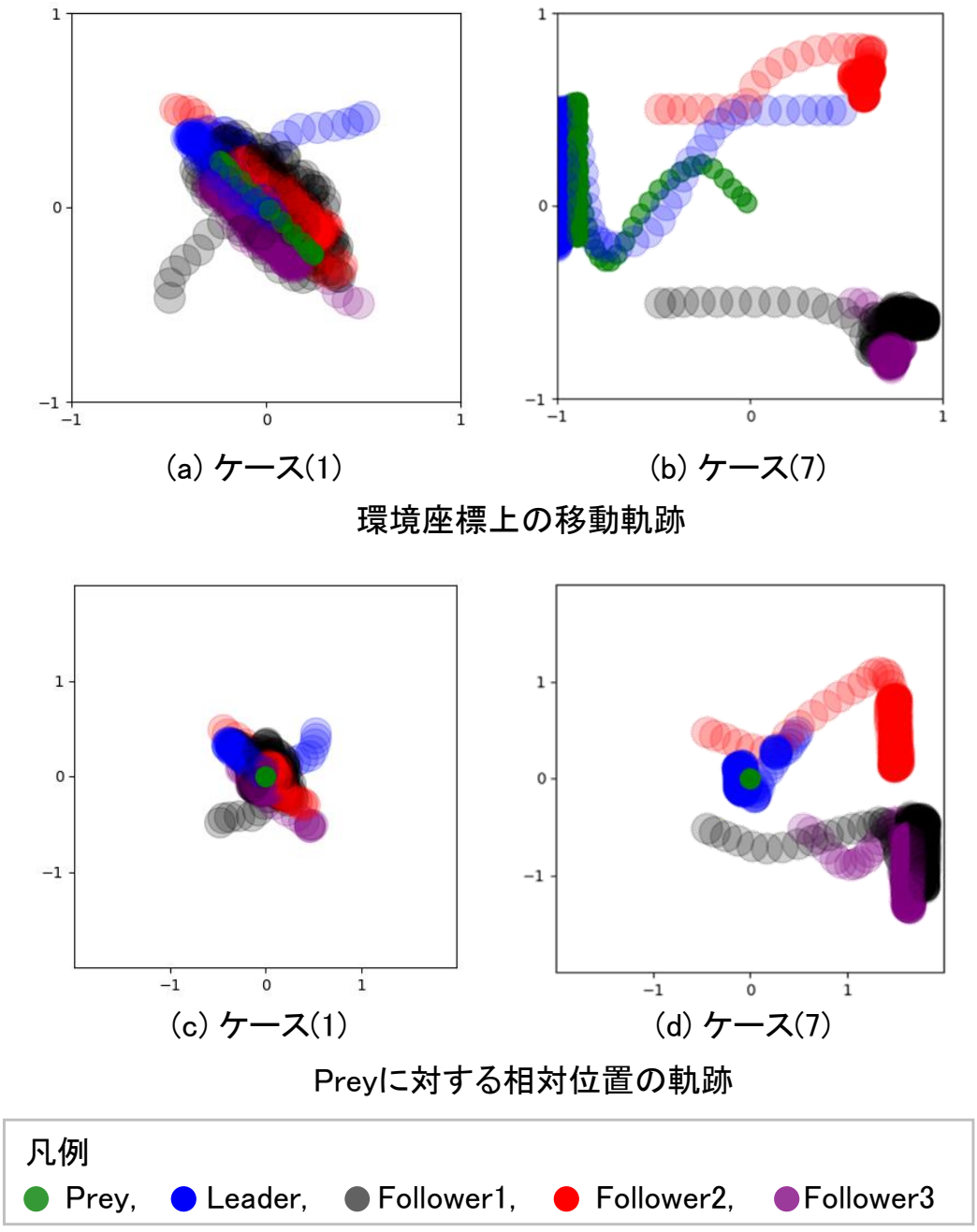


図3.13: 移動軌跡例

3.5.3 実験結果の考察と今後の課題

実験では提案手法と DDPG を組み合わせた場合に最も良い結果が得られた．MADDPG では，Critic が無効な観測情報と行動のセットを得る機会が多くなり学習の妨げになった

可能性が考えられるが，その検証が今後必要である．また，本論文では3.2.2節で設定したように，Leader が常に環境全体を観測可能とし，Follower には観測範囲の制限を課したため，Follower のみが保有する独自の情報が存在しない．Follower 独自の情報が存在しないことにより Follower が自身の独自性を発揮する必要性が相対的に低下し，Leader 強制力が強い効果を持つ一方で，Follower にとって参考情報に過ぎない Leader 指示は有効ではなかったと考える．Follower しか保有しない情報が存在する環境では，Leader が知りえない情報に基づいて行動する必要性が増加するため，Leader 強制力の重要性が低下する．そのため，Leader 指示を参考に Follower が独自性を発揮することが重要になるはずである．エージェント間での保有情報の偏り方と，それを補完することが期待されるエージェント間通信及び強制力の強さのあり方について，今後調査が必要である．チームにとって重要かつ他のエージェントが保有しない情報を持つエージェントがその都度 Leader となって他のエージェントに指示や強制力を与えるチームの組織構造なども考えられる．また学習を通じて Predator チーム内で確立された Leader 指示の内容と，Leader 指示を受けた Follower の行動の間の相関関係や感受性，ケース毎の Leader 指示内容の傾向などは本論文では分析していないため，今後さらなる調査が必要である．

3.6 結言

マルチエージェント，連続空間，部分観測及び競争的環境という困難性の高い特徴を持った追跡問題を対象に，1. Leader-Follower モデルの導入，2. カリキュラム学習という 2 点の提案を行った．Leader-Follower モデルの導入では Leader から Follower に対する一定の強制力を持った通信によって MARL でチームワークを向上する方法を提案した．また，競争的環境において「勝ち方」を教示するため，学習の初期段階において競争的環境にある一方のチームに対しもう一方のチームが「あえて負ける」行動を教示するカリキュラム学習を提案した．

実験では，累積報酬値と Prey の捕捉回数という直接的な結果の比較だけでなく，エージェントの軌跡図を用いた分析も行い，MADDPG が本実験のような部分観測情報環境では固定的な役割分担を持つチームを形成する局所解に陥る一方，提案手法をよりシンプルな学習アルゴリズムである DDPG と組み合わせた場合には提案手法を用いないケースと比べて性能が向上することを確認した．

提案した Leader-Follower モデルとカリキュラム学習の考え方は特定の学習手法に依存せず，本質的にどのような学習手法にも適用が可能であり，今後は Leader と Follower 間の通信内容の調査や学習コストに関する理論的な検討を進めるとともに，提案手法が適する問題設定の特徴を特定し適用先の拡大を行う．また本実験では Prey 捕捉時に全ての Predator

に報酬を与えるいわゆる global reward を用いた．Prey の捕捉は Predator 同士の協力の結果であって，本来であれば貢献度に応じて適切に報酬を分配することが望ましい．そのため次章以降において，エージェントごとの貢献度に応じて協力に対するインセンティブとして適切に報酬を分配するための報酬設計について議論する．

第 4 章

報酬設計へのメカニズムデザインの応用

4.1 序言

本節の構成を次に示す．2 節で提案する VCG の支払いに基づく報酬設計法について述べる．3 節で提案手法を評価するために行う実験について述べ，4 節で実験結果について述べる．さらに，5 節で実験結果を比較するためのベンチマークについて述べる．6 節で実験結果の考察を行い，7 節に本章のまとめを示す．

4.2 VCG の支払いに基づく報酬設計法の提案

本節では，2.7節で述べた VCG メカニズムの考え方に基づき，MARL の報酬関数に応用する方法を提案する．提案する方法では，支払いを計算するために，社会的効用を表す状態価値関数を事前に設計する必要がある．提案する支払いの定義を図4.1に示す．また，あるエージェント A に対する支払いの計算例を図4.2に示す．

図4.1に示すように，エージェント i に課す支払いは状態価値関数で決まる 2 つの値の差で計算する．一つは，エージェント i が環境に存在するときに他のエージェントによって決定される価値の総和であり，図4.1の上のバーが相当する．もう一つは，エージェント i が環境に存在しないときに他のエージェントによって決定される価値の総和であり，図4.1の下のバーが相当する．VCG の支払いと同様，両者の差を評価対象のエージェントが他のエージェントの価値に与えた負の貢献を反映したペナルティとして，評価対象のエージェントが得る報酬から減じる．エージェント i の支払いは，エージェント i が存在することによる他者への影響の程度を表現している．こうした支払いを報酬設計に組み込むことで，エージェントらは利己的な方策を避け，社会的余剰を高めるための効果的な方策を学習する．社会的余剰は MARL におけるエージェントらの目的達成率として定義する．VCG メカニズムは式(2.17)を満たす時，支配戦略誘引両立性（dominant-strategy incentive compatible ：DSIC）が保証される．支配戦略誘因両立性とは，エージェントにとって真の価値を申告す

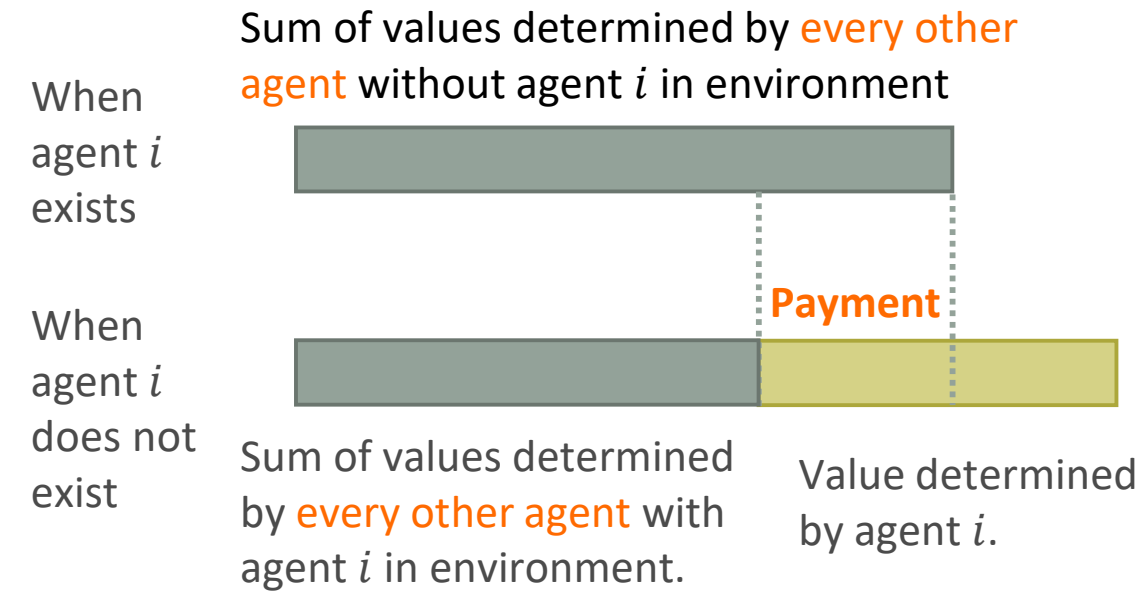


図4.1: The payment definition

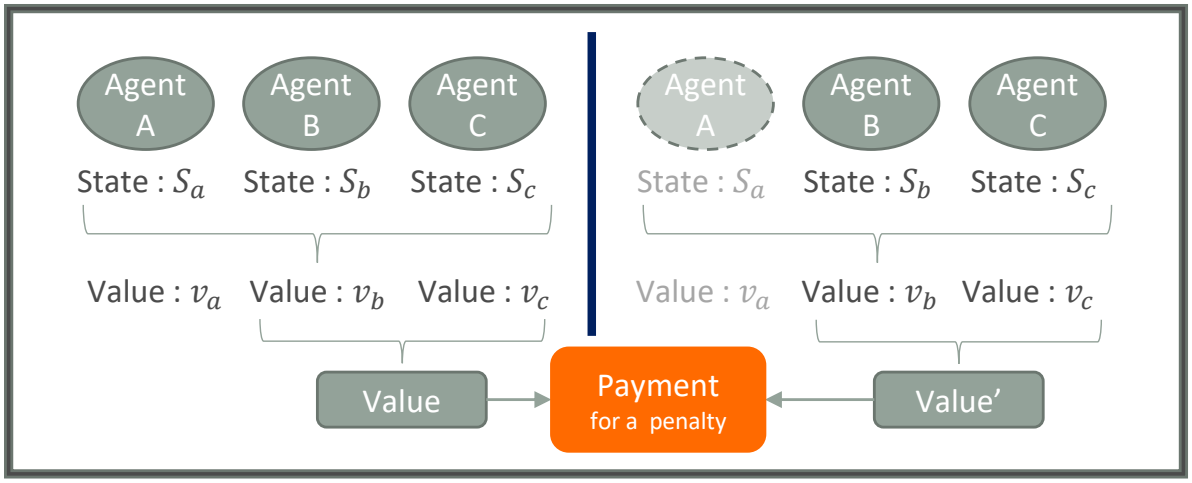


図4.2: Payment calculation example for agent A

ることが最良（支配戦略）となる性質をいう． 一方で，MARL では効率性を満たす決定規則を事前に発見することはできず，このような決定規則を見つけ出すことが MARL の目的である．そこで，MARL においてエージェントが獲得する行動方策を方向付ける教示信号としての報酬を，VCG と同じように社会的余剰を高めるように設計するため，支払い規則

を式(4.1)及び(4.2)として定義する．

$$p_i = \sum_{j \neq i} v_j(S_{-i}) - \sum_{j \neq i} v_j(S) \tag{4.1}$$
$$= V_{-i}(S_{-i}) - V_{-i}(S) \tag{4.2}$$

ここで，状態 S の時の迷惑料を評価するために導入する状態価値 $V(S)$ を事前に定義し，$V(S)$ のうちエージェント i の貢献分を $v_i(S)$，$V(S)$ からエージェント i の貢献分を除いたものを $V_{-i}(S)$ のように表現する．

図4.2はエージェント A に関する報酬設計を表す．左側はエージェント A が存在する環境であり，右側はエージェント A が存在しない環境を示す．また，図中の “Payment for a penalty’’ が式(4.1)及び(4.2)を具体的に表現している．

また，個人の利益と全体の利益が必ずしも一致しない問題を扱うため，支払い額 p_i をエージェント i の報酬の一部として式(4.3)のように定義する．

$$r_i = \alpha R(S_i) - p_i \tag{4.3}$$

ここで，$R(S_i)$ はエージェント i だけの状態で決定される報酬関数 local reward であり，個々のエージェントの価値関数に相当するものとして取り扱う．また α は local reward と支払い額の間の重みを定める係数とする．強化学習を行って知的なエージェントを得ようとする設計者は，パラメータ α によって，local reward と社会的価値のバランスを調整する．

式(4.3)に示すように，報酬関数は準線形に定義されているため，ペナルティの支払いによって，VCG の支払いのような有用なインセンティブとなることが期待できる．本報酬スキームを，Reward Design for MARL based on the Payment Mechanism（RDPM）と呼ぶこととする．

ここで，式(2.15)と式(4.2)の間には，評価対象のエージェントが存在しないことを利用するという共通点があることがわかる．両者の重要な違いは，式(4.2)で評価対象のエージェント i の価値 v_i を含まない V_{-i} のみで決定されるのに対し，式(2.15)は，エージェント i を含むシステム全体を用いることである．

ここで，VCG はワンショットの意思決定においては DSIC のメカニズムを提供するが，現実世界のような動的で反復的な環境を考えると，古典的な VCG では DSIC を保証できない [62, 63]．これは，VCG では長期的な報酬が考慮されていないからである．これに対して，強化学習法の多くは，環境の状態から得られる将来的な期待報酬を考慮して，反復的に目標を達成しようとする．長期的な期待報酬を最大化しようとする MARL の方策あるいは推定価値のあるワンステップの更新において，VCG メカニズムに基づく支払いにより自分

以外の価値に対する考慮が即時報酬として反映されることにより，自身の価値を向上しつつ，協調的な方策あるいは推定価値が強化されることが期待できる．

例えば，Generalized Second Price (GSP) [64] は，一般的に支配戦略の均衡を持たず，真実申告最良のメカニズムでもない．それにもかかわらず，GSP は広く使われており，実際によく機能する．MARL における RDPM も同様であると考えた．RDPM は学習エージェントに社会的に望ましい状態へのインセンティブを与えることができることを，次節に示す実験を通して示す．

4.3 実験

本節では，提案手法を 2 つのシナリオで実験を行って評価する．両方のシナリオともに基本的でシンプルなマルチエージェントタスクであり，提案手法である RDPM を従来手法と比較して評価する．

4.3.1 High-way Driving Problem Domain (HDD)

まず，RDPM を自律走行する学習エージェント間の協調，特に高速道路上での協調に適用する．ここでは問題を単純化し，個々の学習エージェント間の協力という基本的な側面に焦点を当て，高速道路走行問題領域（High-way Driving Problem Domain：HDD）と呼ぶことにする．HDD では，複数の状態を持つエージェントが協力して社会的効用を最大化する必要がある．一般的な道路交通では，全員が初期速度から巡航速度までできるだけ早く速度を上げ，目的地に早く到着して道路の流量を最大化することが望ましい．しかし，各車が他の車の速度に合わせないと危険である．MARL によって協調的に，かつ自発的に加速する独立したエージェント群を得るために，各エージェントにどのような動機付けを与えるかを検討した．本稿で取り扱う HDD では，すべてのエージェントが社会的効用として並走するという同じ目標を共有している．また同時に，個々のエージェントはできるだけ速く走りたいという効用を持っている．そこで，個人の効用を大切にしながら，社会的効用にどれだけ譲歩するかが大きな問題となる．図4.3に，これら HDD 問題の概要を示す．

各エージェント i は，0 から 10 までの速度 s_i を状態として保持し，各時間ステップにおいて，1) Hold：現在の状態と同じ速度を保持する，2) Accelerate：速度を 1 だけ上げる，3) Brake：速度を 1 だけ下げる，の 3 つの行動候補から実際に行う行動を選択する．各エージェントは時間ステップ毎に行動を選択し，遷移後の状態に応じた報酬が与えられる．なお初期状態は 0 から 4 までランダムに初期化する．支払い額を算出するための状態価値関数を次のように定義する．

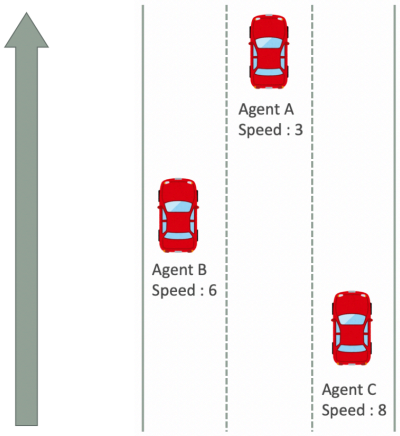


図4.3: The HDD Situation

$$v_i(S) = -\max_{k \in N} |s_k - s_i| \tag{4.4}$$

$$V(S) = \sum_{i \in N} v_i(S) \tag{4.5}$$

HDD の支払額を計算するための状態価値は，各エージェントの速度 s の最大差分の負の総和とする．local reward $R(S_i)$ は，エージェント自身の速度を用いて $R(S_i) = -(s^{\max} - s_i)$ とすることで，速度が遅ければ遅いほど負の報酬を得るように定義する．なお最高速度は $s^{\max} = 10$ とする．このように定義することで，エージェントは自身の速度が高いほど改善する個人の local reward と，状態価値関数で定義される社会的効用の両方を同時に最大化するように学習しなければならない．他の報酬設計と比較するために，各エージェントには次の 3 種類の報酬関数 r_i を使用する．

Global reward (GR):

$$r_i = \frac{\alpha}{c_1} \sum_{j \in N} R(S_j) + \frac{1}{c_1} V(S) \tag{4.6}$$

$$= -\frac{\alpha}{c_1} \sum_{j \in N} (s^{\max} - s_j) - \frac{1}{c_1} \sum_{j \in N} (\max_{k \in N} |s_k - s_j|) \tag{4.7}$$

$$c_1 = n s^{\max} \tag{4.8}$$

Difference rewards (DR):

$$r_i = \frac{\alpha}{c_2} R(S_i) + \frac{1}{c_3} \left(V(S) - V_{-i}(S_{-i}) \right) \tag{4.9}$$

$$= -\frac{\alpha}{c_2} (s^{\max} - s_i) - \frac{1}{c_3} \left(\sum_{j \in N} \max_{k \in N} |s_k - s_j| - \sum_{j \neq i} \max_{k \neq i} |s_k - s_j| \right) \tag{4.10}$$

$$c_2 = s^{\max}, c_3 = n \tag{4.11}$$

提案手法 (RDPM):

$$r_i = \frac{\alpha}{c_2} R(S_i) - \frac{1}{c_3} p_i \tag{4.12}$$

$$= \frac{\alpha}{c_2} R(S_i) - \frac{1}{c_3} \left(V_{-i}(S_{-i}) - V_{-i}(S) \right) \tag{4.13}$$

$$= -\frac{\alpha}{c_2} (s^{\max} - s_i) - \frac{1}{c_3} \left(\sum_{j \neq i} \max_{k \in N} |s_k - s_j| - \sum_{j \neq i} \max_{k \neq i} |s_k - s_j| \right) \tag{4.14}$$

ここで，c_1, c_2, c_3 は報酬をおおよそ ± 1 の範囲内に収めるための正規化係数，n は環境中の学習エージェントの数であり，$n = 5$, $n = 10$, $n = 15$ の 3 つのケースで実験を行う．

HDD では，周囲と同じ速度で走ることは，自身の速度を上げること以上に安全上重要な問題であるため，local reward よりも社会的な効用を重視することとし，α を 0.2 と定義した．3 ケースいずれの報酬設定でも，初期速度からすべてのエージェントがなるべく同期しながら速度を上げ，最高速度に達した場合に最も高い報酬が与えられる．

4.3.2 追跡問題（Predator-Prey Domain：PPD）

追跡問題（Predator-Prey Domain：PPD）[65, 66] は，古典的な協調的かつ競争的なマルチエージェント問題として知られる．

PPD には，Predator（捕食者）と Prey（被食者）という 2 種類のエージェントが存在する．これらのエージェントは，長方形のグリッド上で 1 つの時間ステップで 1 つのセルを移動することができる．Predator の目的は獲物である Prey を捕まえることであり，Prey の目的は Predator から常に距離を確保し，捕食されないことである．4 つの Predator が存在する 9×9 のグリッドの PPD 状況例を図4.4に示す．図4.4における白い四角形が Predator，青い四角形が Prey を示す．今回の実験では，Predator のチームに着目する．PPD の HDD との大きな違いは，Predator チームの学習による効用向上を阻止しようとする相手が同一環境中に存在する点である．Prey は Predator の方策に応答して逃げようと行動する．

Predator も Prey も 1 ステップに 1 セルずつ移動可能という設定のため，1 つの Predator だけでは Prey が正確に逃げ続ける限り，Prey を捕捉することができない．そのため，複数

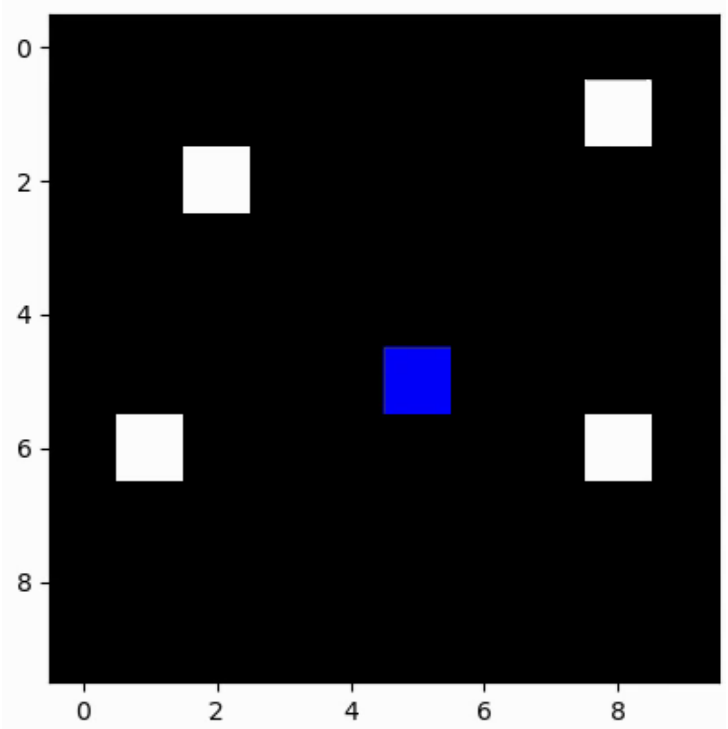


図4.4: PPD situation of 9×9 grid with four predators

の Predator が協力して Prey を捕捉するために協調的に行動する必要がある．Predator は学習方策に基づき，1 ステップごとに左，右，上，下いずれか 1 セル移動するという行動を選択する．

一方，本実験では Prey を事前に設計したスクリプトを用いて行動を選択させる．Prey は，次のステップで移動可能なセルのリストを候補として順に検索し，現時点の全ての Predator の中でユークリッド距離が最近接の Predator の位置から最も遠いセルを選択する．これにより，Prey は最も近い Predator から常に遠ざかろう行動する．移動候補のセルは左右上下の順で検索し，同一距離の場合は先に検索された移動候補を選択する．

次に，Predator の学習に用いる報酬関数は，HDD と同様に global reward (GR)，difference reward (DR)，提案手法 (RDPM) の 3 種類である．global reward は次のように定義する．

GR:

$$r_i = -\frac{1}{c_4} \sum_{j \in N} d_j + \sum_{j \in N} B_j \tag{4.15}$$

ここで，d_i は捕食者 i と獲物の間のユークリッド距離を示す．c_4 は正規化係数であり，グ

リッドサイズ m を用いて $c_4 = (2m+1)\sqrt{2}$ と定義する．N は Predator エージェントの集合であり，B_i は Predator i と Prey が同じ位置にいるときに $+10$ を返し，それ以外では 0 を返す関数とする．

すなわち，global reward で表す PPD における MARL の目的は，全 predator と Prey の間のユークリッド距離を最小化することである．一方，local reward $R(S_i)$ は，エージェント i 自身の Prey との距離を用いて $R(S_i) = -d_i$ と定義することで，Prey に近ければ近いほど良い設定とする．当然，local reward だけでは Predator がチームとして協力せず，各 Predator がそれぞれ Prey に向かって移動しようとするため，協力のためのインセンティブが必要である．

次に，Predator の学習に用いる支払いを計算するための状態価値関数を，Predator の中で Prey との距離が最も大きいものと，自身の Prey との距離の差分として次のように定義する．

$$v_i(S) = -(\max_{k \in N} d_k - d_i) \tag{4.16}$$

このように定義することにより，ある Predator が他の Predator と比べて突出して Prey に接近した場合，課される支払いが増加する．そのため，他の Predator と歩調を合わせて Prey に向かっていくことが期待できる． そのうえで，difference reward と RDPM を以下のように定義する．

DR:

$$r_i = -\frac{1}{c_4}\left\{\sum_{j \in N}\left(\max_{k \in N} d_k - d_j\right) - \sum_{j \neq i}\left(\max_{k \neq i} d_k - d_j\right) + d_i\right\} + B_i \tag{4.17}$$

RDPM:

$$r_i = -\frac{1}{c_4}\left\{\sum_{j \neq i}\left(\max_{k \in N} d_k - d_j\right) - \sum_{j \neq i}\left(\max_{k \neq i} d_k - d_j\right) + d_i\right\} + B_i \tag{4.18}$$

本実験設定では4.3.1節で設定したα は 1 としたので省略する．実験では，$m = 7$ 及び Predator 3 体，$m = 9$ 及び Predator 4 体，$m = 11$ 及び Predator 5 体の 3 つのケースでそれぞれ実験を行った．

4.3.3 学習設定

実験ではいずれのシナリオでも，個々のエージェントの学習アルゴリズムとしてシンプルな Deep Q-Network を使用する．アルゴリズムの概要を Algorithm 2に示す．

ただし，本提案手法は特定のアルゴリズムに制約されることはない．エージェントは自分の状態を含む環境の状態と，自身の行動という 2 種類の情報に基づいてそれぞれの Q ネットワークを学習する．使用する Q ネットワークはシーケンシャルモデルである．ニューラル・ネットワークへの入力は，状態集合で構成される．ニューラルネットワークは 3 段階の隠れ層とし，HDD では各隠れ層はそれぞれ 15，30，15 出力を持つものとする．PPD では同じく 3 層とし，各隠れ層は 80，160，80 出力とする．それぞれの出力は rectified linear unit (ReLU) 関数 [67] によって活性化させる．オプティマイザーは Adam [68] とし，損失関数は Huber loss function [69] とする．学習はそれぞれ表4.1に示す学習パラメータを用いて報酬設定ごとに各 10 回行った．この研究ではハイパーパラメータのチューニングは着目すべき点ではないため，パラメータの調整は少なくとも 1 つの手法が収束するまで調整することにとどめ，各シナリオの実験を通して表4.1に示す同じパラメータを適用した．

Algorithm 2: RDPM algorithm

Initialize replay memory D_i
Initialize Q_i function with random weight
for episode $= 1, \ldots, M$ **do**
  Initialize agent's state S
  for time step $t = 1, \ldots, T$ **do**
    for agent $i = 1, \ldots, n$ **do**
      Select a random action a_i with probability ϵ
      Otherwise select $a_i = \max_{a_i} Q(S_t, a_i)$
      Execute action a_i and change the state
      for agent $j = 1, \ldots, n$ **do**
        if $j \neq i$ **then**
          Evaluate value function $v_j(S)$ and $v_j(S_{-i})$
        end if
      end for
      Calculate payment p_i with Equation (4.1)
      Evaluate reward r_i with Equation (4.3)
      Store transition $(S_{i:t}, a_i, r_i, S_{i:t+1})$ in D_i
    end for
  end for
  reduce ϵ according to the decay setting
  for agent $i = 1, \ldots, n$ **do**
    Sample random minibatch from D_i
    Update action-value function $Q_i(S, a)$
  end for
end for

4.4 結果

4.4.1 HDD (High-way Driving Problem Domain)

HDD の学習時の報酬の推移を図4.5に示す．学習は各 10 回ずつ行い，いずれの報酬設定においても各エピソードにおけるエージェントの平均 global reward（MGR）による評価値を内部的に保持するようにした．MGR は状態に対していずれの報酬設計でも共通に定義される．x 軸はエピソード数，y 軸は各エピソードにおけるエージェントの平均 global reward（MGR）を，指数移動平均（係数 0.99）とその不偏標準偏差をプロットした．報酬関数が異なるため，学習時にエージェントが得た報酬値そのものでは累積報酬を直接比較することができないが，どの報酬関数を用いて学習を行ったかに関わらず，式(4.6)及び式(4.7)で表さ

表4.1: 学習パラメータの設定

	HDD	PPD
Number of learning agents	5, 10, 15	3, 4, 5
Number of episodes	10,000	10,000
Steps per episode	30	50
Learning rate	1.0×10^{-3}	1.0×10^{-4}
Memory size	10,000	20,000
Batch size	64	64
Gamma	0.99	0.99
Initial Epsilon value	0.9	0.9
Epsilon decay	rate at 0.9997	linearly decreasing at 0.0001 and minimum at 0

れる MGR の値は，学習エージェントによるチームタスクの達成度として捉えることができ，MGR の値を用いて異なる報酬関数間の結果を比較することができる．(a)，(b)，(c) の結果を比較すると，global reward を学習の報酬関数として使用した場合，個々のエージェントに適切なフィードバックを与えることができないため，学習エージェントの数が増えるにつれて学習が不安定になっていることがわかる．DR は GR よりも良い結果を示したが，RDPM は常に最良となっている．

(b) と (c) では，MGR の値が −10 付近になったときに RDPM の報酬の増加率が一

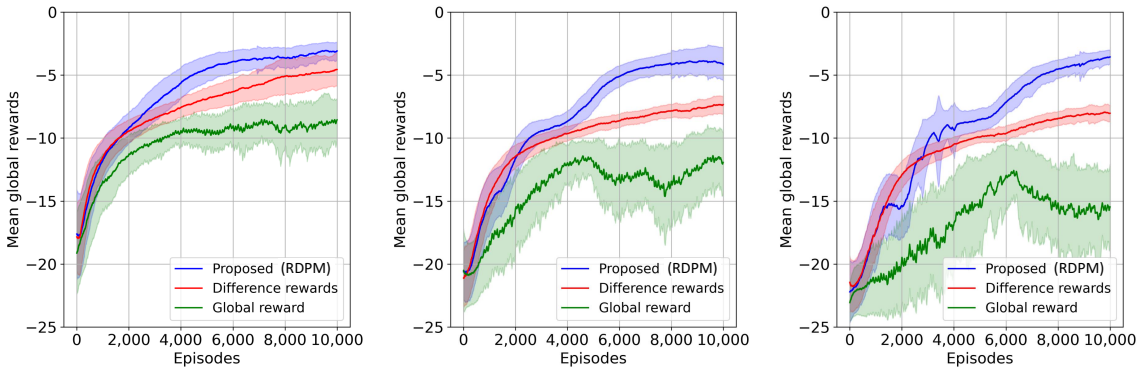
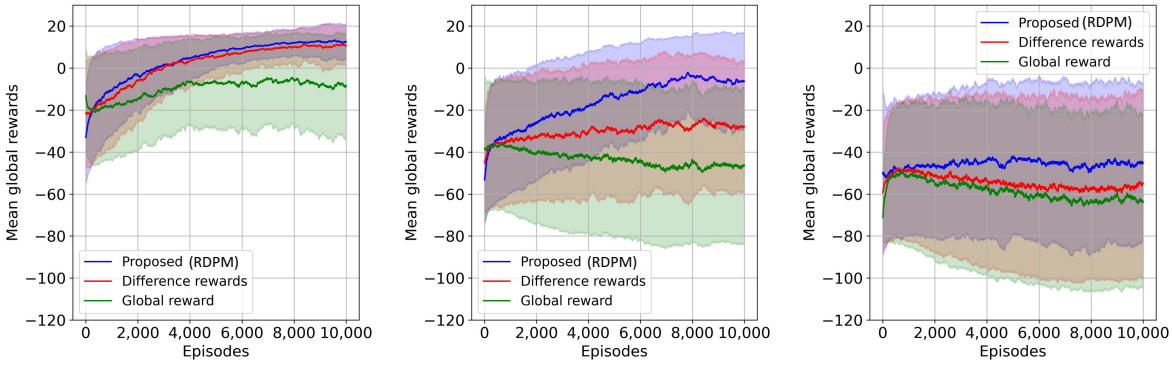


図4.5: Reward transitions during training of HDD



(a) Grid size 7x7, 3 agents (b) Grid size 9x9, 4 agents (c) Grid size 11x11, 5 agents

図4.6: Reward transitions during training of PPD

時的に減少したように見え，difference reward ではこの値付近に収束している．これは，MGR= −10 のあたりに何らかの「壁」があって，このようなこぶができたのかもしれない．この点について，4.5.3節において，追加調査する．

これらの結果は，local reward と支払い額の間の重みを定める係数 α を 0.2 に設定した場合のものである．HDD では，周囲と同じ速度で走ることは，自身の速度を上げること以上に安全上重要な問題であるため，local reward よりも社会的な効用を重視する設定として，1 よりも小さい値を設定した． α を 0.2 よりも大きくして 1 に近い値にすると ，local reward の評価の度合いが大きくなる．本実験では local reward は自分の速度だけで計算されるため，エージェントにとっては2.6節で述べたように自身の報酬を増加させる学習が容易になるが，社会的効用を増加させることが容易になるわけではない．α が 0.2 以外の場合の影響は，4.5.4節において追加調査する．

4.4.2 PPD (Predator-Prey Domain)

PPD の学習時の報酬の推移を図4.6に示す．各軸の定義とスムージングの方法は図4.5と同様である．

図4.6の (a) では，difference reward と RDPM は標準偏差の分布が狭く，両者ともに近い値に収束するのに対し，global reward は分布が広く低い値に収束している．これは，difference reward と RDPM で学習したエージェントは同程度の性能を発揮し，global reward で学習したエージェントは difference reward 及び RDPM と比べて性能が悪いことを示している．(b) では，difference reward と global reward の結果は (a) と比べて悪化し，一方で RDPM は依然として MGR が増加している．(c) では，どの報酬設計も学習が進ん

でいないが，これも依然として RDPM が最も MGR が高い値に収束している．difference reward と RDPM の差は，問題の複雑さ（エージェントの数とグリッド数）が増すにつれて，より明確となっていることが分かる．

4.5 ベンチマーク

4.5.1 HDD (High-way Driving Problem Domain)

本節では，HDD における学習した結果より，得られたエージェントの性能を比較評価する．本節における評価時は，エージェントの初期速度は学習時のようにランダムに決定するのではなくエージェント毎に 0 から 4 の順に設定（エージェント 1 の初期速度は 0，エージェント 2 の初期速度は 1，エージェント 3 の初期速度は 2，$\cdots$ といったように）し，ベンチマークテストとして global reward，difference reward，RDPM の各報酬設計で各 10 回ずつ学習した結果得られた，各 10 種類 $\times$ 3 種類の報酬設計による学習済み方策を用いて 30 ステップの HDD を実行した．表4.2に，ベンチマークテストの結果得られた MGR の値と標準偏差を示す．また，表4.3に，difference reward と global reward のベンチマーク結果の RDPM に対する T 検定時の p 値を示す．T 検定は，まず比較する対象群間に対し F 検定を行い，2 群間が不等分散ではないことが示唆された場合に Student の T 検定を，それ以外の場合に Welch [70] の T 検定を用いた．結果，すべてのケースで MGR 値に有意な差が確認された．global reward の場合の学習は，図4.5で示したように発散傾向であり，ベンチマークテストの結果としても 3 つの報酬設計の中で常に最悪値となった．difference reward では MGR の値は global reward より常に高い値となり，RDPM の場合はさらにいずれの報酬設計の場合よりも MGR の値が高くなった．

本節で示したベンチマークは MGR の値だけで比較しており，それぞれの報酬設計によって得られた個々の学習エージェントが具体的にどのようなふるまいを行った結果，本節の結果となっているかが分かりにくい．そのため，学習済みエージェントがどのようなふるまいを行っているかを観察する追加のベンチマークを行う．追加のベンチマークテストの結果は，4.5.3節で述べる．

4.5.2 PPD (Predator-Prey Domain)

本節では，HDD と同様に PPD についても学習した結果得られたエージェントの性能を比較評価する．表4.4に PPD のベンチマーク結果を示す．global reward, difference reward, RDPM の各報酬設計で各 10 回ずつ学習した結果得られた，各 10 種類 $\times$ 3 種類の報酬設計による学習済み方策を用いてそれぞれ 1,000 回のエピソードを行い，捕食者が獲物の捕獲に

表4.2: Benchmark results of HDD

Reward design	$n = 5$ MGR (Std.)	$n = 10$ MGR (Std.)	$n = 15$ MGR (Std.)
RDPM	-3.44 (0.52)	-4.61 (1.34)	-3.44 (0.42)
DR	-4.95 (1.10)	-7.21 (0.85)	-8.18 (0.78)
GR	-8.48 (1.20)	-11.75 (2.08)	-15.44 (4.00)

表4.3: T test p-values of HDD benchmark

Reward design	$n = 5$ RDPM	$n = 10$ RDPM	$n = 15$ RDPM
DR	1.77×10^{-3} **1	6.11×10^{-5} **2	1.55×10^{-12} **1
GR	3.06×10^{-8} **1	3.52×10^{-8} **2	4.97×10^{-6} **2

**1 :$p < .01$ with Welch's t test

**2 :$p < .01$ with Student's t test

表4.4: Benchmark results of PPD

Reward design	$n = 3, m = 7$ Success rate (%) Mean (Std.)	$n = 4, m = 9$ Success rate (%) Mean (Std.)	$n = 5, m = 11$ Success rate (%) Mean (Std.)
RDPM	97.20 (4.64)	76.66 (15.08)	31.91 (5.41)
DR	95.98 (2.70)	44.47 (14.95)	30.51 (4.32)
GR	65.03 (11.74)	27.00 (2.64)	23.77 (2.65)

成功したエピソードの数を成功率とした．また，表4.5に DR と GR のベンチマーク結果の RDPM に対する T 検定時の p 値を示す．表4.4の各エージェント数 n 及びグリッド数 m 毎の RDPM，DR，GR の各成功率を比べると，どのエージェント数及びグリッド数においても，RDPM が常に最も高い成功率となっている．表4.5では，DR の $n = 5,\ m = 11$ の場合を除いた他のすべてのケースで成功率に差が確認できた．また RDPM で学習した Predator は，Prey をはさむなどの協調行動をとることが観察された．$n = 5,\ m = 11$ のときの結果では RDPM でも約 32% の成功率でしかないが，それでも常に最良の結果となっている．

表4.5: T test p-values of PPD benchmark

Reward	$n = 3, m = 7$	$n = 4, m = 9$	$n = 5, m = 11$
design	RDPM	RDPM	RDPM
DR	4.82×10^{-1} *1	1.45×10^{-4} **1	5.31×10^{-1}
GR	4.01×10^{-6} **2	1.82×10^{-6} **2	8.98×10^{-4} **2

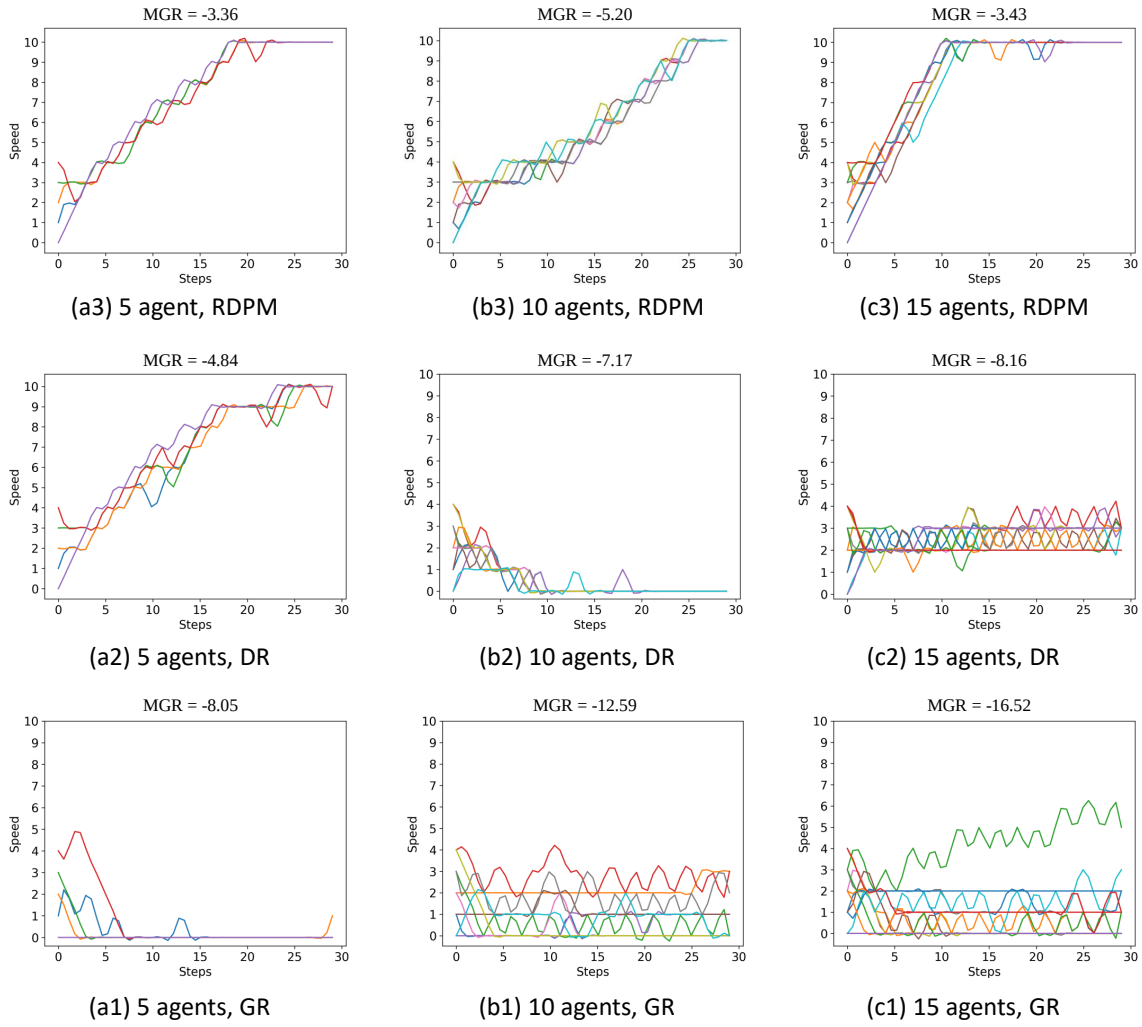
*1 :$p < .05$ with Student's t test
**1 :$p < .01$ with Student's t test
**2 :$p < .01$ with Welch's t test

4.5.3 HDD におけるエージェント速度の推移の分析（追加ベンチマーク）

4.5.1節において MGR の値による統計的比較を行った．本節では，それぞれの報酬設計によって得られた個々の学習エージェントが具体的にどのようなふるまいを行っているかを確認する．実験では， 3 つの報酬設計法と 3 種類のエージェント数（$n = 5, 10, 15$）それぞれの設定で 各 10 回学習した結果，$3 \times 3 \times 10 = 90$ 個の学習済み方策が得られている．比較すべき方策の数が多いため，本節では報酬設計法とエージェント数の違いによる典型的な傾向により差異を確認する． 各 10 回の学習の結果得られた方策のうち，平均値に最も近い MGR 値を達成したケースを「典型的な」方策として選択した．典型的なケースにおけるエージェントの速度推移を図4.7に示す．

(a1)，(b1)，(c1) に示すように，GR を用いた場合，エージェントは協調して速度を上げることができなかったことがわかる．特に (a1) では，全エージェントが最も低い速度に合わせるような行動が獲得されている．すべてのエージェントが速度 0 になったとき，式(4.6)の第二項（状態価値関数に基づく支払いに相当）は最大の 0 になるが，第一項（local reward 相当）は最小になる．つまり，全員が他の速度に合わせることを重視するあまり，local reward を放棄したといえる．速度を合わせるにあたって，学習時もエージェントの初期速度は 0 から 4 の間でランダムに初期化されるため，最大速度である 10 を目指すよりも最下端である 0 に合わせる方が容易であったと考えられる．(b1) と (c1) に示すように，エージェントの数が増えると MGR はさらに悪化し，エージェントは自分の速度を適切に調整できなくなっている．図4.5に示したように GR による学習は発散傾向であり，ベンチマーク結果についてもうまく協調できていないことを示している．

DR を使用した場合，(a2) ではエージェントが同期して同時に速度を上げることに成功していることがわかる．しかし，(b2) では，(a1) と同じように速度を 0 に調整している．また，(c2) では，エージェント間での速度の調整と増加に苦労していることがわかる．(a1)，(b1)，(c1)，(b2)，(c2) は，いわば式の第一項を放棄するような局所最適であり，このよう



（注：いくつかの線は重複している）

図4.7: Speed transitions of agents in HDD

な局所最適が図4.5の RDPM を用いた学習時の報酬推移において観察された，MGR= −10 付近で発生していたコブの原因になっていると考えられる．

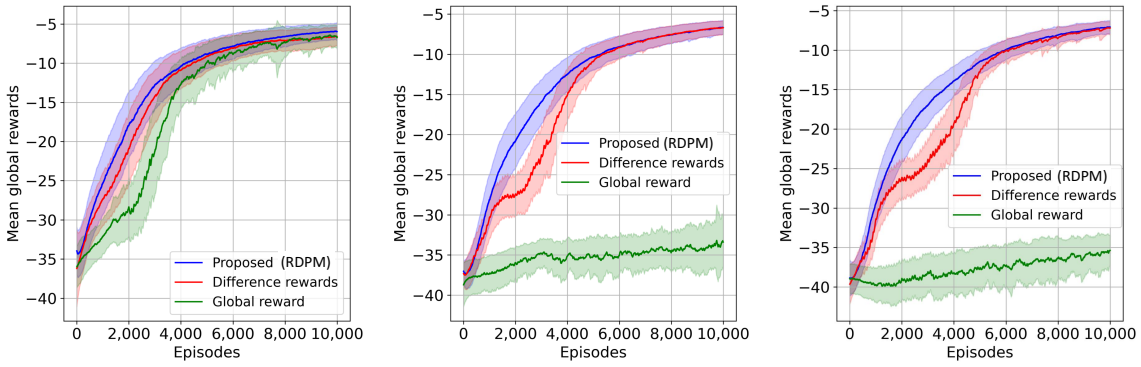
これに対して，RDPM を用いた場合は (a3)，(b3) 及び (c3) で示すように，エージェントは互いに他と協調的に速度を上げることができている．例えば，あるステップにおいて他よりも速度の速いエージェントは，Hold（現在の速度と同じ速度を保持する）や Brake（速度を 1 だけ下げる）という行動を選択しても local reward としては低い報酬しか得られない．にもかかわらず，RDPM の結果では，あるステップにおいて速度の速いエージェントが一時的に Hold や Brake を選択して，遅いエージェントに合流を図っている様子が観察で

きる．すなわち，他より速いエージェントは支払いのペナルティを減らすために，一時的に個々の報酬である local reward を犠牲にしたといえる．このことから，提案した手法を用いることで協調的な方策の獲得に成功したといえる．

4.5.4 HDD における α の変更例

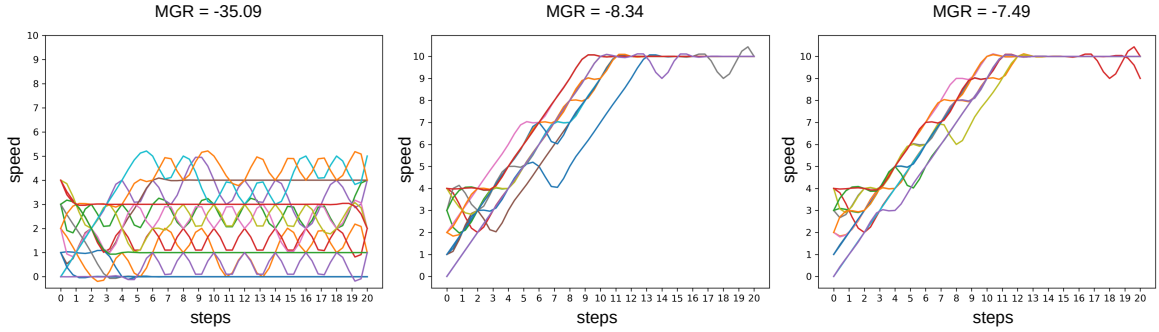
4.3.1節に示した HDD の実験では，安全性を反映した社会的効用を重視した設定として α を 0.2 に設定した．本節では，個人的効用と社会的効用の間のバランスを調整するための重みであるα の効果を検証するために，α の値を変えて追加実験を行う．最も極端な例として，個人的効用をより重視する設定としてα を 1.0 に設定し，$n = 5, 10, 15$ の 3 ケースについてそれぞれ 3 種類の報酬設計で各 10 回の学習を実施した．図4.8に，$\alpha = 1.0$ と設定した場合の HDD の学習時の報酬遷移を示す．α が大きくなると負の報酬を得る可能性が高くなるため，MGR の値（縦軸）は図4.5に示した $\alpha = 0.2$ の場合の結果と比較することはできない．図4.8に示したように，GR ではエージェント数の増加に伴い，安定した学習ができないのは図4.5と同様である．一方，DR と RDPM では，図4.5と比較して (a)，(b)，(c) いずれにおいても報酬が安定的に増加しており，学習が容易になったと考えられる．ただし，この結果は $\alpha = 1.0$ で学習したエージェントが $\alpha = 0.2$ で学習したエージェントよりも優れているということを意味しない．

4.5.3節と同様に，$\alpha = 1.0$ で学習した結果得られた学習方策のうち，典型的なケースにおけるエージェントの速度推移を図4.9に示す．典型的とは，各 10 回の学習の結果得られた方策のうち平均値に最も近い MGR 値を達成したケースを指す．$n = 15$ の GR の比較として図4.9の (a) と図4.7の (c1)，同 DR の比較として図4.9の (b) と図4.7の (c2)，及び同 RDPM の比較として図4.9の (c) と図4.7の (c3)，それぞれの α を変更したことによる影響を観察すると，明らかに DR の場合が $\alpha = 1.0$ で学習した図4.9の (b) の方が改善している．このように，α によって学習結果が影響を受けることは確かであるが，α が大きければ良いということではない．HDD では，α を大きくすると local reward を重視することとなってエージェントが加速するモチベーションが高まり，結果として最大速度で走るという望ましい状態が得られやすくなることが考えられる．しかし，他の問題では必ずしも個人の利益としての local reward を重視することが良い結果に向かうとは限らない．社会的に望ましい行動を見つけることの難しさは問題ごとに異なる．個人の利益を重視して学習の収束が良くなる効果と，社会的な利益を重視することによる学習難化の効果のバランスを取るような結果を予測するのは困難である．現時点でのベストプラクティスは，設計者が MARL によって解決を図ろうとする対象問題において，個人的利益と社会的利益をどの程度重視するかに応じて α を設定することである．



(a) 5 agents (b) 10 agents (c) 15 agents

図4.8: Reward transitions during training of HDD with $\alpha = 1.0$



(a) 15 agents, GR (b) 15 agents, DR (c) 15 agents, RDPM

(Some lines overlap.)

図4.9: Speed transitions of agents in HDD with $\alpha = 1.0$

4.6 実験結果の考察

HDD と PPD の 2 つのシナリオで実験を行い，提案手法はいずれにおいても従来手法と比べて良い結果となった．本節では，結果全体を俯瞰するとともに，さらなる潜在的な問題点について考察する．

4.6.1 全般

GR，DR 及び RDPM それぞれの報酬設計の違いによって，学習エージェントが具体的にどのように報酬を得て，どのように行動を強化するように作用しているかを具体例を用い

表4.6: Reward example in HDD

No.	speed of agents					reward for agent 0		
	0	1	2	3	4	GR	DR	RDPM
1	10	10	10	10	10	0.00	0.00	0.00
2	9	10	10	10	10	**-0.10**	-1.02	-0.82
3	10	9	10	10	10	**-0.10**	**-0.20**	**0.00**

て議論する．表4.6に，$\alpha = 0.2$ とした場合の 5 エージェントでの HDD 問題における各エージェントの状態 s（すなわち各エージェントの速度）と，その状態においてあるエージェントが各報酬設計でどのような報酬を得るかの例を示す．表4.6の No.1 のケースでは，全てのエージェントが最高速度である 10 で走っている状態であり，この場合は個人的効用と社会的効用の両方が最大化されるため，どの報酬設計でも 0.00 が与えられる．HDD における 1 ステップでの報酬は，どの報酬設計でも 0.00 が最大値である．

No.2 の設定はエージェント 0 のみが速度 9 で，その他が 10 のケースであり，No.3 の設定はエージェント 1 のみが速度 9，その他が 10 のケースである．GR の場合は，No.2 のように自分だけが遅い状態でも，No.3 のように自分以外であるエージェント 1 だけが遅い状態でも，-0.10 という同じ報酬を得る．GR ではエージェント 0 にとって自分の報酬を向上させるために何をすべきかを学び取ることが難しいことがわかる．

これに対して，No.2 の設定における DR と RDPM は，一人だけ他より遅い速度であるエージェント 0 が非難されるような大きい負の報酬を与えることになり，報酬がエージェント 0 にとって望ましくない状態であることの教示信号となっていることがわかる．DR と RDPM の違いが生じる例として，No.3 の状態のような状況が挙げられる．No.3 では，RDPM では 0.00 が与えられるのに対し，DR では小さな負の報酬となっている．No.3 では，エージェント 1 のみが他より遅い速度であり，エージェント 0 が負の報酬を得るいわれはないが，DR ではエージェント 0 以外のエージェントのために社会的効用が減少する場合でも負の報酬が与えられることがある．DR の支払いには，対象となるエージェントが受け取る社会的効用による価値の一部が含まれているためである．

DR は全てのエージェントの価値と，対象エージェントが存在しなかった場合の価値の差分で評価することから，前者に対象エージェントを評価に含むこととなる．一方 RDPM は，対象エージェントが存在する場合の対象エージェント以外の価値と，対象エージェントが存在しない場合の対象エージェント以外の価値の差分で評価するため，支払いに対象エージェントの要素を含まない．このため，local reward と組み合わせ，他者を考慮し社会的効用への貢献度を評価する支払いメカニズムとして利用する場合，DR よりも RDPM が優れてい

ると考える．しかし，DR を用いる場合は必ずしも local reward と組み合わせて支払いとして利用することが必要ではないため，直接的に DR を個人効用と社会的効用を同時に評価するような実装方法も考えられる．このように DR による報酬を定義した場合，報酬の構成が全く異なるため，提案手法と直接的な比較はできなくなるが，どの程度の性能が得られるかを示すことで，本問題に対するより深い理解が得られる可能性が考えられる．

4.6.2 課題

本節では，提案した報酬設計法及び実験結果における課題について議論する．

まず第一に，VCG は，動的で反復的な環境における長期的効用を考慮しておらず，支配戦略誘因両立性（DSIC）を保証していないことが挙げられる．同様に，VCG は限定合理的なエージェントを仮定していない．MARL における学習エージェントは，少なくとも学習段階では完璧に合理的な方策を持っているわけではない（だからこそ学習している）．MARL における RDPM の理論的な限界についての証明が期待される．

2 点目は，定数 α を用いた点である．α は個人的効用と社会的効用の間のバランスを調整するための重みを定義しており，α の設定は重要であるが，本研究では手動でア・プリオリに設定した．本来であればこの設定は自動化されるべきと考える．

3 点目は，支払いを計算するために設定した状態価値関数についてである．式(4.1)及び式(4.2)に示したように，エージェント i の貢献分を除いた状態価値 $V_{-i}(S)$ を容易に計算可能な場合でなければ，RDPM を使用することが困難であるという問題がある．例えば，あるタスクを実行するために従属的な部分作業を多数のエージェントが協力して行わなければならない場合，あるエージェントが存在しなかった場合を想定することが難しくなり，そのエージェントの貢献度を評価することは困難である．この制約は DR など，評価対象のエージェントの不在性を仮定する他の手法にも共通する問題であり，提案手法固有の問題ではないが，エージェント間の協調を促進したい場合に用いるべき報酬設計において，この制約は非常に強い制約となる．多くの問題領域では，あるエージェントが存在しなかった場合を仮定して評価することは容易ではなく，本節で議論する課題の中でも最も強い制約と考える．このため，次章ではこの制約を緩和することを議論する．

4 点目は計算コストの問題である．VCG メカニズムは最適値を計算する必要があるため，複雑な問題ではその計算コストによって実質的に実行不可能になることがある．支払い計算の構造が複雑になればなるほど，計算コストも増大する．したがって，計算コストの緩和と効率性の厳格さのバランスをとることが実用上重要である．本研究では，エージェント間の協力関係に着目し，速度や距離などのエージェントの集合的な測定値である最大値を用いて v を定義した．集団の同一性を評価するためには，すべてのエージェントの状態値を毎回計

測し支払いを計算するするのではなく，その最大値や最小値で抽象化した計測値を用いれば，計算コストを削減することができる．HDD や PPD における社会的効用の定義を振り返ると，HDD では他のエージェントの速度と合わせることであり，PPD では Predator チームが歩調を合わせて Prey との距離を詰めていくことであり，いずれも協調エージェント間の同一性によって状態価値関数を定義したため，速度や距離のエージェント内の最大値と自身の速度や距離との差分で評価が可能であった．ただし，こういった計算コストの緩和方法は一般化できるわけではない．

4.7 結言

本章では，Vickrey-Clarke-Groves (VCG) メカニズムによる支払いアルゴリズムをベースに，社会的効用に基づくペナルティを反映した報酬をマルチエージェント強化学習 (MARL) に適用する，新しい報酬設計法 (RDPM) を提案した．提案する手法では，VCG と同様に，他のエージェントの評価に対する負の貢献度を反映した支払いを定義した．個々のエージェントには，個人の行動のみで評価される local reward と，支払いに基づいて評価されるペナルティとしてのネガティブな報酬からなる報酬を与えた．この手法を highway driving problem domain（HDD）と predator-prey domain（PPD）の 2 つのシナリオに適用して実験を行った．HDD では，RDPM が global reward (GR) や difference reward (DR) よりも優れていることを示した．比較的単純な問題領域である HDD では，DR と RDPM の両方が学習を収束に導くことができたが，GR は 学習が不安定となった．さらに，RDPM で学習したエージェントは，ベンチマークテストで協力することに成功していることを示した． また，HDD よりも複雑な PPD 問題領域では，RDPM を報酬に適用することで，GR や DR を適用するよりも高いスコアが得られることを実証した．しかし，4.6.2節で述べたように，いくつかの課題が残されている．特に，評価対象のエージェントの不在性を仮定する RDPM や DR では，あるエージェントが存在しなかった場合を容易に仮定して評価できる必要があり，協調問題を取り扱う場合には特に強い制約となる．そのため次章では，この制約を緩和する方法を検討する．

第 5 章

メカニズムデザインを応用した報酬設計の適用可能性向上

5.1 序言

本章の 2 節では，4章で提案した VCG の支払いに基づく報酬設計法の適用可能性向上の必要性について述べる．3 節では，提案手法の適用可能性を向上するために仮想的に追加のシミュレーションを行うことを提案する．4 節では新たな提案手法を評価するための実験内容について述べ，議論する．5 節では本節の内容をまとめ，今後の研究について述べる．

5.2 適用可能性向上の必要性

4章において，VCG メカニズムによる支払いアルゴリズムをベースに，社会的効用に基づくペナルティを反映した報酬をマルチエージェント強化学習 (MARL) に適用する，新しい報酬設計法 (RDPM) を提案した．VCG の支払いは，対象となるエージェントが存在する場合と存在しない場合について，他のエージェントが決定した値の合計の差によって，エージェントの貢献度を測定する．また類似の手法として，2.6節で紹介した difference reward (DR) にも共通する課題として，対象となるエージェントが存在しないことをどのように仮定するかという問題がある．これらの手法は，一般に報酬が例えば $R(S) = R(s_i) + R(S_{-i})$ のように，各エージェントの個々の状態の和や積で容易に評価できる問題を扱っており，エージェントの不在性を即座に評価できる問題に特化している．この仮定は，エージェント間の協調を扱う際に非常に強い制約となる．なぜなら，協調タスクは通常，複数のエージェントが依存する部分的な作業を同時に分割して処理することが要求されるため，エージェントが存在しない場合の状態評価は単純な和や積では表現できないからである．

そこで本章では，実際に学習しようとしている対象環境に加えて，エージェント数が 1 つ少ない仮想的な環境を用意し，$n-1$ エージェントの環境と実際に学習しようとしている n

エージェントの環境の間のエージェントの本質的な価値評価の差に基づいて報酬を計算する方法を提案する．便宜上，仮想的な環境を「仮想環境」，解決しようとする環境を「現実環境」と呼ぶこととする．

なお，本章では最もシンプルな学習手法であり difference reward の文献 [42] でも使用されているQ 学習に焦点を当てて議論する．

5.3 仮想環境を用いた VCG の支払いに基づく報酬

本節では，2.3.2節で述べた Q 学習について，環境中における学習エージェントが複数であることを前提に再整理する．強化学習では，エージェント i の目的は，式(5.1)のようにマルコフ決定過程において将来の割引された期待報酬の合計を最大化するような方策 π_i を見つけることである．

$$v_i(S, \pi_i) = \sum_{t=0}^{\infty} \gamma^t G(r_t^i | \pi_i, S_0 = S) \tag{5.1}$$

ここで，S_0 は初期状態，r_t^i は時間 t におけるエージェント i の報酬，$\gamma \in [0, 1)$ は割引率である．$v(S, \pi)$ は状態 S における戦略 π のもとでの価値と呼ばれる．Q 学習では，状態 S における行動 $a^i \in A_i$ の「価値」，すなわち行動価値関数を以下のような Q 関数と呼ぶ．

$$Q_i^*(S, a^i) = r(S, a^i) + \gamma \sum_{S'} p(S'|S, a^i) v(S', \pi^*) \tag{5.2}$$

ここで，A は行動候補の集合，$p(S'|S, a^i)$ はエージェント i が行動 a^i を行った時に，状態が S から S' に遷移する確率である．Bellman Equation は次のように表される．

$$Q_{i,t+1}(S_t, a_t^i) \leftarrow (1 - \alpha) Q_{i,t}(S_t, a_t^i) + \alpha \left(r_t^i + \gamma \max_{a^i} Q_{i,t}(S_{t+1}, a^i) \right) \tag{5.3}$$

ここで，α は学習率を表す．

ここで，4章で検討したように，メカニズムデザイン (MD) の考え方を報酬設計に応用する方法を検討する． 経済学では，投資家が消費した結果得られる満足度を効用と呼ぶ．準線形の効用関数を仮定すると，2.7節で述べたように効用 u は，得られる「価値」と消費され

る「支払い」に応じて，$u = v - p$ と表される．今，式(2.19)を式(5.3)の報酬 r_{t+1}^i に適用しようと試みる．その際に考慮すべき最も重要な MARL と MD における概念の違いが，「価値」と「効用」の定義であると考える．MARL を含む上位概念である強化学習では，無限回の更新を仮定して長期的に期待される報酬を割引いたものを「価値」としているのに対し，経済学における効用は，1 回の支払いによって消費することで得られる「価値」で構成されている．そのため，式(2.19)示した支払いを強化学習に適用する際には，ある時間ステップ t で得られる即時報酬 r_t^i をエージェントの効用として扱うこととした．

MD では，エージェントが $u_i(\theta_i, a) = v_i(\theta_i, a) - p_i$ で示される準線形効用選好を持ち，式(2.19)で示される支払いを与えられた場合，$u_i(\theta_i) \geq u_i(\hat{\theta}_i)$ であることが証明されている．ここで，θ_i はエージェント i の真のタイプ，$\hat{\theta}_i$ はエージェント i が申告したタイプを示す．すなわち，エージェント i は真のタイプである θ_i を申告した際にもっとも良い効用を得ることができる．

これを強化学習の観点から検討する．エージェントの価値の総和が最大になったときに最高の報酬がエージェント i に与えられるならば，エージェント i は学習を通じてそれを達成するために強化を行うはずである．ここで，状態 S_t においてエージェントが行動集合（共同行動と呼ぶ）$\mathbb{A}_t = \{a_t^1 \in A, \ldots, a_t^n \in A\}$ を選択し，「現実環境」で計算できる即時の任意の報酬 $R(S_t, \mathbb{A}_t)$ が与えられると仮定する．そして，エージェント i の即時報酬関数 r_t^i を，状態 S_t と共同行動 $\mathbb{A}_t$ に応じた支払い p_i をエージェントに課すことで，次のように準線形の報酬に変換する．

$$r_t^i(S_t, \mathbb{A}_t) \stackrel{\text{def}}{=} R_t(S_t, \mathbb{A}_t) - p_i \tag{5.4}$$

もし，共同行動の別の候補が $\hat{\mathbb{A}}_t$ であるとき，$r_t^i(S_t, \mathbb{A}_t) \geq r_t^i(S_t, \hat{\mathbb{A}}_t)$ が成り立つならば，エージェントに $\hat{\mathbb{A}}_t$ よりも $\mathbb{A}_t$ を選択するようにインセンティブを与えることができる．MD の考え方を素直に適用すると，決定ルール g は次のようになる．

$$g_t(S_t) = \underset{\hat{\mathbb{A}}_t}{\operatorname{argmax}} \sum_{i \in N} R_t(S_t, \hat{\mathbb{A}}_t) \tag{5.5}$$

そして，支払いのルールは次のようになる．

$$p_i = \sum_{j \neq i} R_t\left(S_t, g_t(S_t^{-i})\right) - \sum_{j \neq i} R_t\left(S_t, g_t(S_t)\right) \tag{5.6}$$

しかし，この場合 $\sum_{j\neq i} R_t\left(S_t, g_t(S_t^{-i})\right)$ と $\sum_{j\neq i} R_t\left(S_t, g_t(S_t)\right)$ を計算するためには，すべての時間ステップにおいて可能なすべての共同行動の候補となる組みあわせを網羅的に計算する必要があり，計算コストが高く実用的な環境ではほとんど実行不可能である．

また，この仮定では MD の $v_t(\theta_i, g(\theta))$ を暗黙的に $R_t\left(S_t, g_t(S_t)\right)$ と解釈している．MD における価値 v は，MARL ではどのような意味を持つのだろうか．MD では，エージェントは確固たる価値観とそれに基づく選好を持っていると仮定し，その判断に社会的な観点からペナルティを与えることを考えるが，MARL の観点からは，エージェントの価値観は学習によって育成され，学習によって変化する．その過程で教示信号として与えるのが報酬である．そこで本提案では，MD の観点での価値を，MARL において 2 種類に分けて考える．一つは，環境によって決定され，エージェントに与えられる即時報酬 R であり，もう一つは，エージェントがその時点で持っている内的な価値基準，すなわち将来の期待値である Q である．そして，環境から決まる v の代わりに即時報酬 R を使い，Q を使って，他のエージェントへの影響を考慮しながら，ペナルティに相当する支払い額を次のように計算することを提案する．

$$g_t(S_t) = \underset{a\in A}{\operatorname{argmax}} \sum_{i\in N} Q_i(S_t, a_t) \tag{5.7}$$

$$p_i = \sum_{j\neq i} Q_j(S_t^{-i}, g_t(S_t^{-i}) - \sum_{j\neq i} Q_j(S_t, g(S_t)) \tag{5.8}$$

2.4節で述べたように，ここでは完全に分権的で独立したエージェントを仮定しているため，$i \neq j$ であれば Q_i は他の Q_j に依存しない．したがって，式(5.8)は次のように表すことができる．

$$p_i = \sum_{j\neq i} \max_{a_t\in A} Q_j^{-i}(S_t^{-i}, a_t) - \sum_{j\neq i} \max_{a_t\in A} Q_j(S_t, a_t) \tag{5.9}$$

そして，仮想的な $(n-1)$ 環境を作成し，$Q_j^{-i}(S_t^{-i}, a_t)$ を計算する．本提案では，シンプルに $(n-1)$ 個のエージェントを持つ環境を n 個作成することとする．これらの環境を「仮想」と呼ぶこととする．これらの「仮想環境」では，エージェント数以外の学習パラメータは「現実環境」と同じとし，エージェントは global reward (GR) を得ながら同時に学習する．全体の流れは図5.1に示すとおりであり，Q 更新の手順は図5.2に示すとおりである．ま

た，アルゴリズムを Algorithm 3に示す．本提案手法を，“Penalty based on the Payment mechanism using Minus-One environment” (PPMO) と呼ぶこととする．

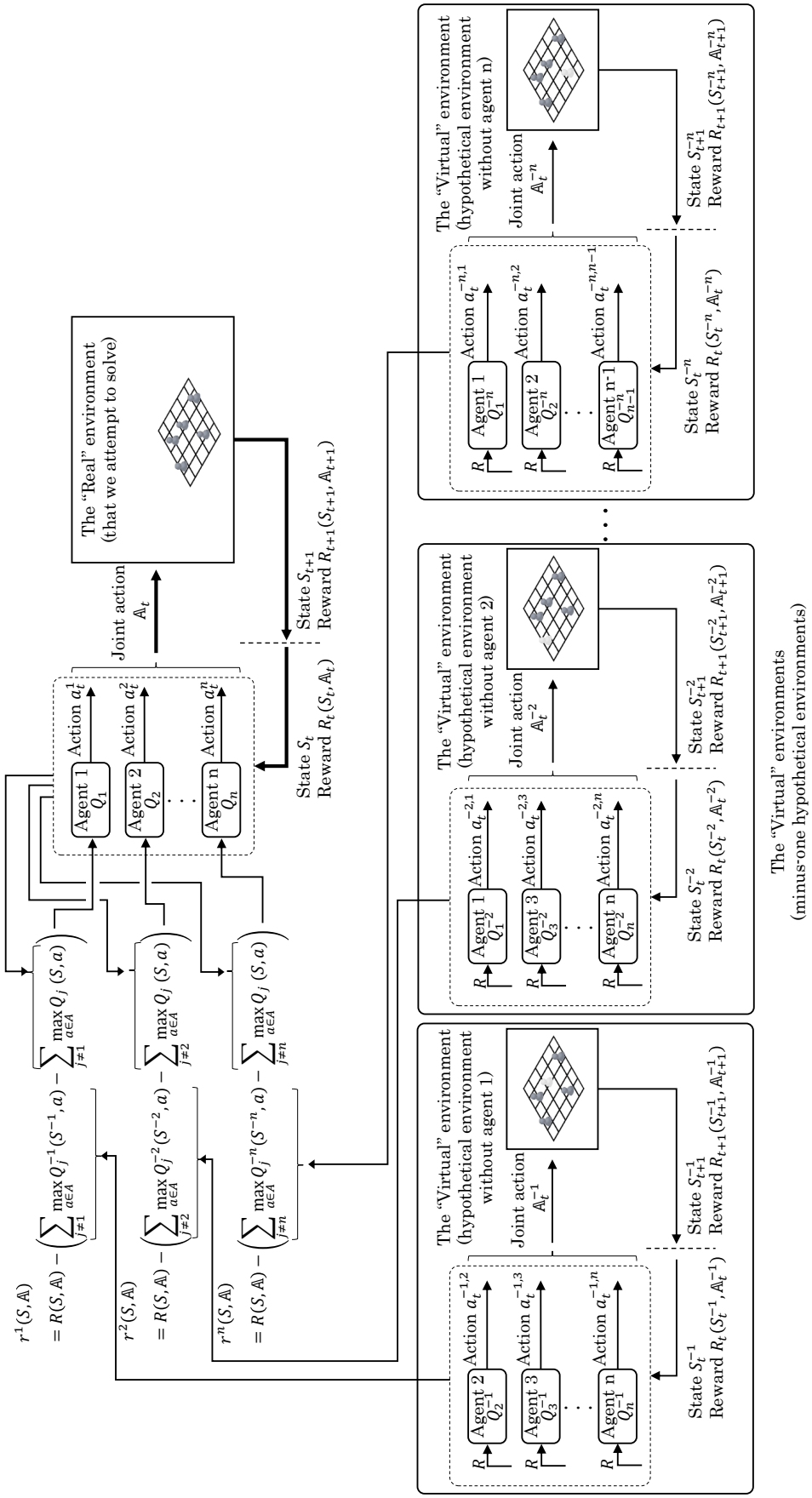


図5.1: The system of PPMO

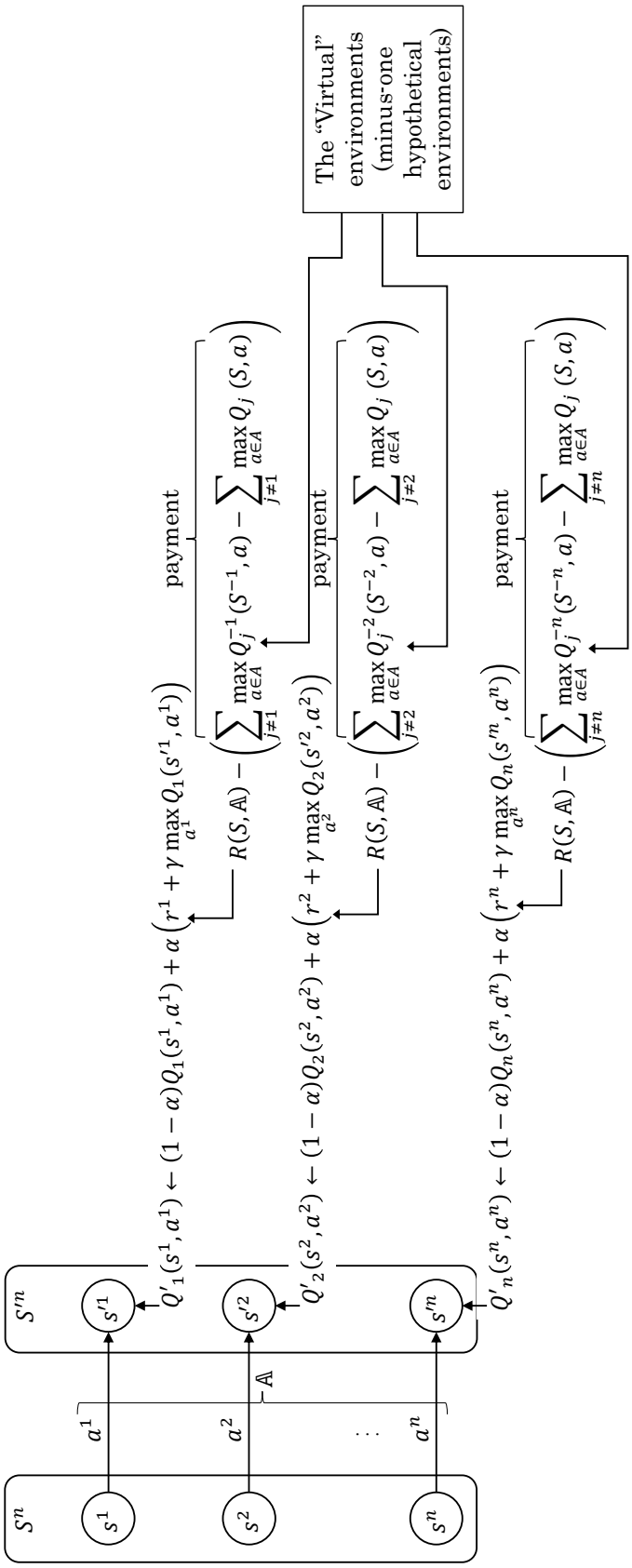


図5.2: The Q update procedures

Algorithm 3: PPMO algorithm

Make "real" environment with n agents $i \in N$ and initialize Q_i function with random weight
for agent $i = 1, n$ **do**
  Make "virtual" environments with $n-1$ agents $\{j \in N | j \neq i\}$ and set $Q_j^{-i} = Q_j$
end for
for episode $= 1, M$ **do**
  Initialize agents' state S
  for *timestep* $t = 1, T$ **do**
    for agent $i = 1, n$ **do**
      Select a random action a^i with probability ϵ Otherwise select $a^i = \max_{a \in A} Q_i(s_t, a)$
      for agent $j \in N, j \neq i$ **do**
        Select a random action a^j with probability ϵ Otherwise select $a^j = \max_{a \in A} Q_j^{-i}(s_t^{-i}, a)$
      end for
    end for
    Execute all actions and change all states
    for agent $i = 1, n$ **do**
      Calculate reward r_i for "real" environment with Equation (5.4) and (5.9) and update Q_i
      for agent $j \in n, j \neq i$ **do**
        Calculate reward r_j^{-i} for "virtual" environment with Equation (2.13) and update Q_j^{-i}
      end for
    end for
  end for
end for

提案手法は，他のエージェントの Q を考慮して自分の Q を更新するという点で，Nash Q-learning [71] と類似しているが，ナッシュ均衡解を直接求める必要がない点が最も大きな相違点である．厳密な意味での効率性 $r_t^i(S_t, \mathbb{A}_t) \geq r_t^i(S_t, \hat{\mathbb{A}}_t)$ は保証されないが，外部性に基づいて計算された支払いは，エージェントが他者と協力することを促すと期待できる．

5.4 実験

本節では，potential-based difference rewards (PBRS) とその他の複数の報酬設計を比較検証するために使用された 2 つのシナリオ [42] で提案手法である PPMO を実験を通じて評価する．どちらのシナリオも基本的でシンプルなマルチエージェントタスクであり，他の報酬設計と比較して PPMO のタスク達成度の観点から評価する．

5.4.1 Grid world domain (GWD)

PPMO がどのように機能するかを示し，他の手法と性能を比較するために GWD の実験を行う．GWD では，複数のエージェントが 2 次元のグリッド上に分散したPoint Of Interest（POI）を観察しようと試みる．この問題の目標は，各エージェントができるだけ重複なく異なる POI の観測を学習することである．エージェントは最初，グリッドの中央付近に配置され，各エージェント $i \in N$ は，時間ステップごとに {上に 1 セル移動，下に 1 セル移動，左に 1 セル移動，右に 1 セル移動，移動せず現在の位置を保持}，の行動候補から行動を選択する．POI を観測する品質はエージェントが POI に近づくほど高くなるが，全てのエージェントの中で最も高い式(5.10)で与えられる品質の観測のみが社会的な効用としてカウントされる．GWD におけるエージェントの初期配置と POI の位置は図5.3に示すとおりである．

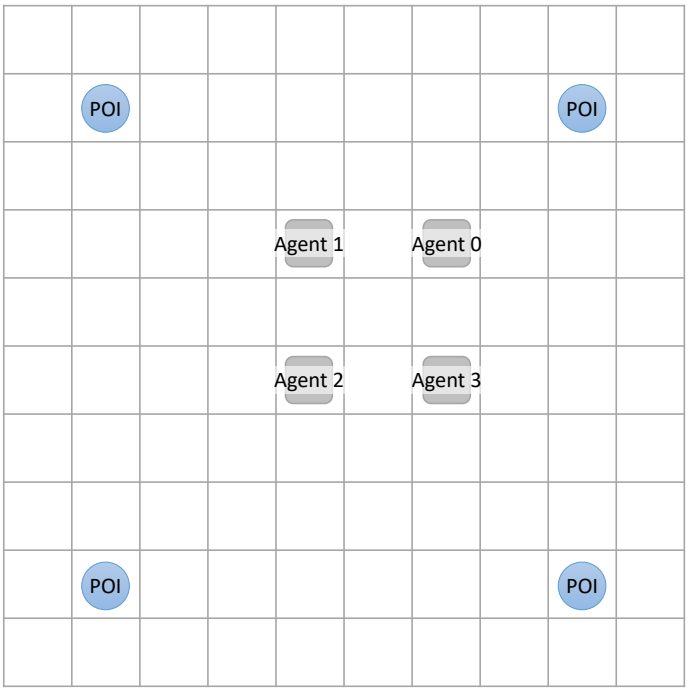


図5.3: Initial settings of Grid World Problem

各 POI を観測することによる品質 O は，次式で与える．

$$O(agent, poi) = \begin{cases} value(poi) & dist(agent, poi) < 2 \\ \frac{value(poi)}{dist(agent,poi)^2} & 2 <= dist(agent, poi) <= 10 \\ 0 & \text{otherwise} \end{cases} \tag{5.10}$$

ここで文献 [42] では，品質 O は以下の通り定義されている．

$$O(agent, poi) = \begin{cases} value(poi) & dist(agent, poi) <= 2 \\ \frac{value(poi)}{dist(agent,poi)^2} & 2 < dist(agent, poi) <= 10 \\ 0 & \text{otherwise} \end{cases} \tag{5.11}$$

しかし，この設定では得られる結果が文献 [42] に記載された結果とかなり異なり，得られる累積の global reward がかなり高くなる．式 (5.10)によって得られる結果は，文献 [42] に記載された結果とかなり近い結果を再現可能であるため，文献 [42] に記載の式(5.11)が誤記であると判断し，式(5.10)を採用する[*2]．

エージェント i の local reward を L_i，社会的効用としての global reward を G，difference reward を D_i とし，POI の観測品質 O に基づいて，以下のように定義する．

$$L_i(t) = \sum_{poi \in P} O(i, poi) \tag{5.12}$$

$$G(t) = \sum_{poi \in P} \max_{i \in N} O(i, poi) \tag{5.13}$$

$$D_i(t) = G(t) - \sum_{poi \in P} \max_{i \in N_{-i}} O(i, poi) \tag{5.14}$$

文献 [42] では，Potential-based reward shaping（PBRS）についても検証している．PBRS では，典型的な適用方法は次のようになる．

$$R(S, a, S') = G(S, a, S') + \gamma \Phi(S') - \Phi(S) \tag{5.15}$$

PBRS では，設計者が問題領域に固有の事前知識を用いて手動で $\Phi(S)$ を実装する必要がある．文献 [42] では，difference reward におけるエージェントの不在性をポテンシャルとして用いて PBRS として適用することを提案しており，これを Counterfactual as Potential (CaP) と呼ぶ．CaP のポテンシャル関数は次のように定義される．

$$\Phi(s) = \sum_{poi \in P} \max_{i \in N_{-i}} O(i, poi) \tag{5.16}$$

[*2]なお本件について著者に連絡を取って確認したところ，すでに所属大学を離れており実装コードにアクセスできないため確定的なことは言えないが，結果はよく再現されているとの回答を得ている．（“I think your reproduced results look impressively similar”）

difference reward，global reward 及び local reward に対して PBRS 手法を適用した報酬設計をそれぞれ，$DRiP$，$G + ManualPBRS$，及び$L + ManualPBRS$ と呼ぶ．いずれも予め，問題領域に固有の事前知識を用いて設計する必要のあるヒューリスティックを利用して報酬を成形するもので，次のように与えられる．

$$\Phi(s) = dist(agent, center) \tag{5.17}$$

ここで $center$ は，初期化の際にエージェントが配置される，grid world の中心セルの位置を示す．式(5.17)は，エージェントが初期位置からどの程度離れているかをポテンシャルとして定義することで，エージェントが初期位置から離れるように促し，従って自然と POI に近づくことを促進する効果をもたらすものである．このようなドメイン固有の知識を報酬に反映することは，エージェントが効果的な方法で学習するのに大いに役立つ．しかし，POI がエージェントの初期位置から離れたグリッドの端に位置していることが前提となっていることからも明らかなように，一般性は失われてしまう．

本実験において提案する PPMO は，L を式(5.4)における R として使用し，次のように定義する．

$$r_i^{\mathrm{PPMO}}(t) = L_i(t) - p_i. \tag{5.18}$$

本実験では，文献 [42] の “Figure 5: 10×10 GridWorld Domain” に示される設定と同様に設定した．すなわち，50 ステップを 1 エピソードとして 2,500 エピソード，$\alpha = 0.1$，$alpha\_decay\_rate = 0.9999$，$\epsilon = 0.2$，$epsilon\_decay\_rate = 0.9999$，$\gamma = 0.9$，$num\_agents = 4$，及び $num\_POIs = 4$ とする．各 POI の観測品質 O は最大で 1 とする．ただし，実行開始時に 4 つのうち 1 つの POI をランダムに選び，例外として観測品質の最大値を 5 に設定する．ランダムに決定する，ある POI を観測することによって得られる価値を他と異なる値に設定することで，ヒューリスティックを事前に設計することを困難にしている．次節で述べる学習結果では，各タイムステップにおける社会的効用 G の合計と，30 回の統計的学習試行に基づく平均の標準誤差を表すエラーバーを含む結果を示す．誤差は，$\frac{\sigma}{\sqrt{T}}$ として計算する．σ は標準偏差，T は統計的学習試行回数を表す．

5.4.2 GWD の実験結果

GWD の実験結果を図5.4に示す．GWD では，各 POI に最も近いエージェント以外の残りのエージェントは，その POI に関して global reward，すなわち社会的効用 G に影響を与えない．つまり，G を自身の報酬として受け取る学習エージェントは，自分が POI から近い順に 2 番目以降の場合，自分がどう行動すれば G を増加させることができるのかを見極めることが難しくなり，結果として学習が困難となる．すなわち，G ではそもそも POI に向かって移動することが良いということを学習することが難しい．一方，local reward L の場合は，どの行動が自分の報酬 L を増やすことができるかを識別することは容易である．GWD では，個々のエージェントが他のエージェントを考慮することなく貪欲に同じ POI に向かって移動した場合，G がその分増加することはないが，一方で負の影響もない．少なくとも，L で学習する場合，POI に向かって移動する行動は容易に強化されるため，ある一定程度までは G を増加させる行動が獲得される．その結果，L は G よりも結果として社会的効用である G を高めやすく，図5.4のように L の方が G よりも高いglobal reward を得ている．

PBRS 法では，G に PBRS を適用した $G + ManualPBRS$，L に PBRS を適用した $L + ManualPBRS$，D に PBRS を適用した $DRiP$ が，それぞれ PBRS を用いていない場合よりも高い global reward を得る結果となった．$G + CaP$ についても，G よりも高い global reward を得ている．これらの結果は，文献 [42] の図 5 に示された結果と類似しており，文献の内容を再現できていてるといえる．文献における実験を再現できていることを確認した上で，提案手法について議論する．図5.4における提案手法 $PPMO$ の結果は，$DRiP$ に次いで 2 番目に高い global reward 値を発揮し，$L + ManualPBRS$ がそれに続く．なお，$DRiP$ と $L + ManualPBRS$ は，いずれもドメイン固有の事前知識に基づくポテンシャル関数を使用しており，本質的に提案手法よりも有利である一方，事前知識を用いない提案手法が同程度の global reward を発揮している．

さらに追加の実験として，エージェント数増加の影響を調査する．エージェントの数を 4 から 20 に増やた結果を図5.5に示す．GWD でグリッドサイズを変えずにエージェント数を増やした場合，エージェントの密度が高くなることによって，自然といずれかのエージェントが POI の近くにいる状態になりやすくなる．図5.5と図5.4とを比較すると，すべての方式で図5.4よりも図5.5のほうが global reward の値が高くなっているのはそのためである．一方で，エージェント数が増えると，エージェントが仮想的にいないことを利用した difference reward の効果は相対的に小さくなる．これは，ドメイン固有の事前知識を利用せず，difference reward をポテンシャルとして利用する $G + CaP$ が G とほぼ同じであり，$DRiP$ と D の差が小さくなることからも確認できる．特筆すべきは，ドメイン固有の知識

も PBRS 法も用いない $PPMO$ がトップグループの成績を示したことであり，提案手法がエージェント数の多いマルチエージェント環境で有利であることを示唆している．

5.4.3 Beach problem domain (BPD)

2 つ目の問題領域では，文献 [42] において Beach problem domain（BPD）と呼ぶ環境を用いる．BPD も GWD と同様に社会的効用を最大化するためにエージェントが協調しなければならない問題であり，混雑問題と呼ばれる [72]．BPD では，砂浜（ビーチ）に沿ってホテルが立ち並ぶような場面を想定し，各ホテルから出発する客（エージェント）が，ビーチを複数に分割したセクションのうち，どのセクションで一日を過ごすかを選択しなければならないという状況を想定する．エージェントは，最初にビーチのある一つのセクションからスタートし，現在のセクションの隣のセクション（左または右）に移動するか，じっとしているかのいずれかの行動を選択する．報酬は，そのセクションにいるエージェントの数と，1 セクションあたりの最適な収容エージェント数 ψ との差に応じて与えられる．1 セクションにいるエージェントの数が最適な収容エージェント数に近ければ近いほど望ましい状態のため，この差は，小さいほど良い．この問題の難しい点は，環境中に存在するエージェントの数が $\psi * \ sections$ よりも圧倒的に多く設定されることで，ある 1 つのセクションを除き，他のすべてのセクションに最適な数のエージェントがいて，それらのセクションから溢れた残りの全てのエージェントが残された最後のセクションに集中しているときに社会的効用が最も高くなるところにある．すなわち，エージェントの効用の合計はシステム全体の効用とはならず，むしろ，大多数のエージェントが不幸になるような社会の形が望ましい状態として定義される．この問題設定では，local reward L は，そのエージェントが存在するビーチセクションに居合わせたエージェント数で決まり，次のように定義される．

$$L(s,t) = x_t e^{\frac{-x_t}{\psi}} \tag{5.19}$$

ここで，$L(s,t)$ はビーチセクション s のローカル報酬，x_t はタイムステップ t でそのビーチセクションにいるエージェントの数，ψ は 1 セクションあたりの最適な収容エージェント数，B はビーチのセクションの集合である．社会的効用を示す global reward G は次のように定義される．明らかに，local reward で学習したエージェントは，社会的効用を示す global reward G を増加させるようには全く有効に機能しない．

$$G(t) = \sum_{s \in B} L(s,t) \tag{5.20}$$

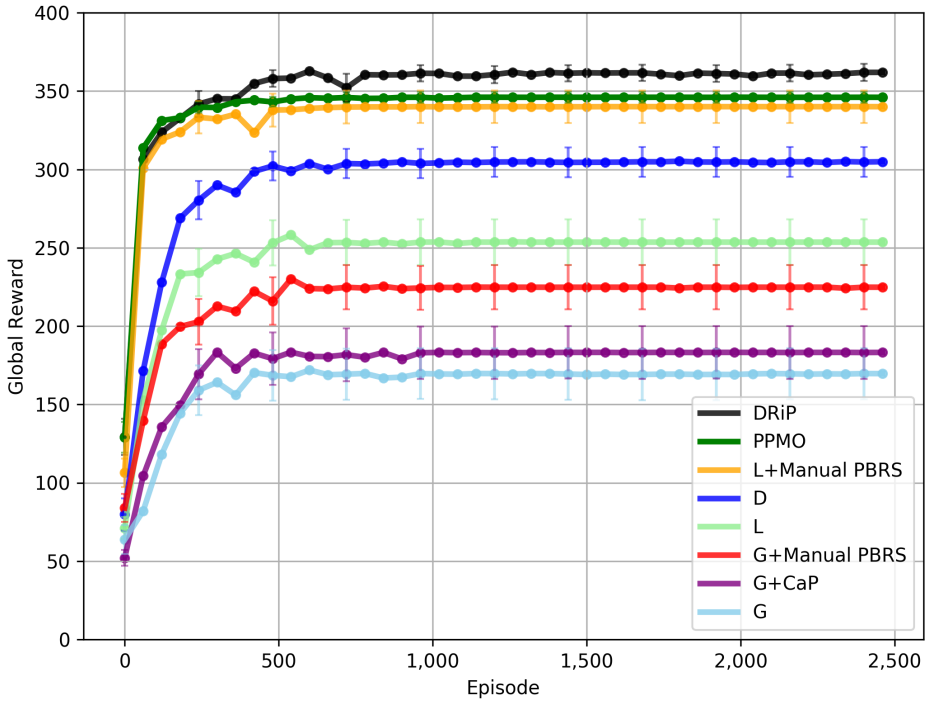


図5.4: 10×10 GWD with 4 agents

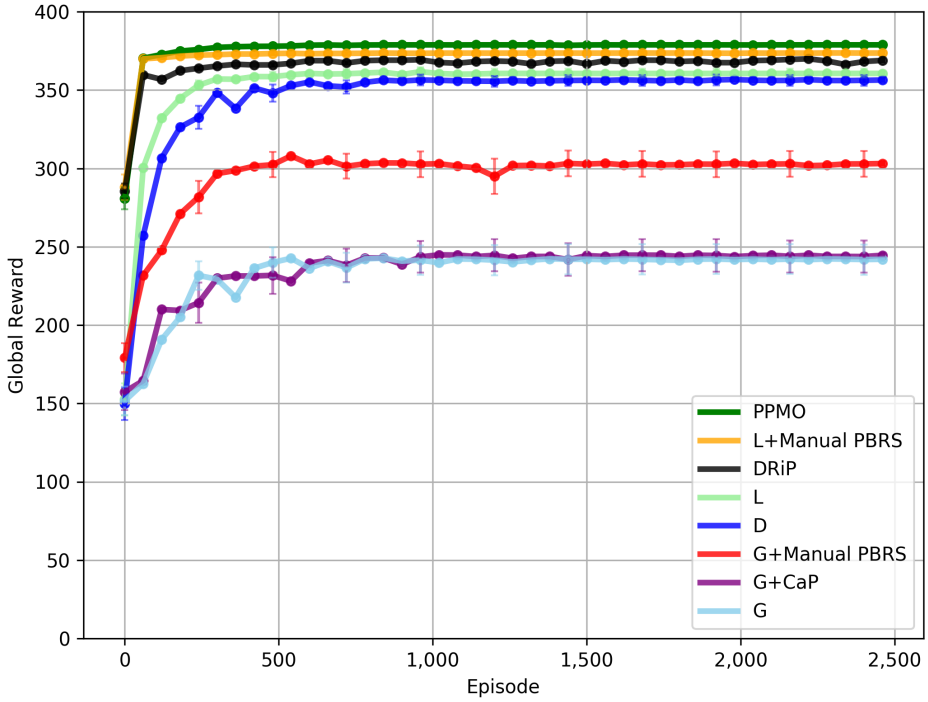


図5.5: 10×10 GWD with 20 agents

difference reward D は次式で与えられる．

$$D_i(t) = L(s,t) - (x_t - 1)e^{-\frac{x_t - 1}{\psi}} \quad (5.21)$$ [*3]

事前知識に基づいて人が設計する必要のあるポテンシャル関数 $\Phi(s)$ は，以下の場合以外では 0 として定義する [42]．

$$\begin{aligned}
&\text{if } agent_id \in [0, \psi - 1] && \Phi(0) = 10 && (5.25)\\
&\text{if } agent_id \in [\psi, 2\psi - 1] && \Phi(1) = 10 && (5.26)\\
&\text{if } agent_id \in [2\psi, X - 2\psi - 1] && \Phi(2) = 10 && (5.27)\\
&\text{if } agent_id \in [X - 2\psi, X - \psi - 1] && \Phi(3) = 10 && (5.28)\\
&\text{if } agent_id \in [X - \psi, X - 1] && \Phi(4) = 10 && (5.29)\\
&\text{if } otherwise && \Phi(4) = 10 && (5.30)
\end{aligned}$$

ここで，$X = \sum_{i=0}^{size(B)} x_i$ はエージェントの数を示す．この各 PBRS 手法と $DRiP$ で使用されるヒューリスティクスは，問題領域に固有の事前知識に基づいて，自身のエージェント ID によって 5 つのビーチのセクションのうち，生来的にどのセクションに居るべきかを教示する信号であり，間接的に問題の解答を与えているに等しい．1.2節で述べたように，そもそも学習によってエージェントの行動を獲得することの目的は，

- エージェントが活動する環境に未知の要素があり，エージェントが遭遇する全ての状況を設計者が予め予想することができない状態空間への対応

[*3] 文献 [42] では，D は次のように定義されている．

$$D_i(t) = L(s,t) - (x_t - 1)e^{\frac{-x_t - 1}{\psi}} \quad (5.22)$$

しかし，式(2.15)の difference reward の定義によれば，D は次式の通りとなるはずである．

$$\begin{aligned}
D_i(t) =& G(S,t) - G(S_{-i},t) && (5.23)\\
=& \sum_{s \in B} L(s,t) - \sum_{s \in B} L(s_{-i},t) && (5.24)
\end{aligned}$$

エージェント i が存在しないセクションについては，第 1 項と第 2 項が同じとなるため，差分なく相殺される．従って，difference reward は式(5.21)となるはずであり，文献 [42] における D の定義が誤記であると判断した．結果がよく再現されていることも，式(5.21)の定義を裏付けている．また，文献 [42] では，本文中に $end\_episode = 20,000$ と記載されているが，BPD の学習結果を示す図ではすべて 10,000 エピソードまでしか表示されていない．本件論文では，$PPMO$ が 10,000 エピソードでは収束していないように見受けられたこともあり，文献 [42] の本文の記載にあわせて，すべて 20,000 エピソードまで学習を行い，結果を示した．

- 望ましい結果を規定した場合でも結果への過程解法をプログラムとして実現するのが困難な問題への対応

という点にある．事前知識として解法が与えられるのであれば，そもそも学習を用いる必要がないと考えられる．また，問題領域に固有の事前知識に基づく報酬は，その問題領域でのみ有効な教示信号として働くため， 得られる方策の一般性は失われる．

本問題領域では L がよい学習信号として機能しないことから，本実験における PPMO では，次のように式(5.4)における R として G を利用する．

$$r_i^{\mathrm{PPMO}}(t) = G(t) - p_i. \tag{5.31}$$

実験は，文献 [42] の “Figure 1: Single-step Beach Domain Results’’ に示される設定と同一とした．具体的には, $\alpha = 0.1$, $alpha\_decay\_rate = 0.9999$, $\epsilon = 0.05$, $epsilon\_decay\_rate = 0.9999$, $\gamma = 0.9, num\_agents = 100$, beach sections $size(B) = 5$, そして 1 セクションあたりの最適な収容エージェント数は $\psi = 7$ とする．この条件において，1 ステップの行動選択を行う学習を 20,000 エピソード実施する．ビーチセクションは 5 つに分かれ，左からセクション 0，1 の順で，一番右側をセクション 4 とする．エージェントの初期位置は，エージェント ID が $num\_agents/2$ 以下のエージェントはセクション 1（左から 2 番目），残りのエージェントはセクション 3（右から 2 番目）とする．

5.4.4 BPD の実験結果

BPD の実験結果を図5.6に示す．

図5.6の L はエピソードが進むにつれて社会的効用を示す global reward の値が低下しており，良い学習信号として機能していないことが分かる．$G + ManualPBRS$ と $G + CaP$ は G よりも高い global reward 値となっており，$DRiP$ と D は，GWD と同様，他の手法よりも高い global reward となっている．事前知識に依存しない D が非常に良い global reward 値となっていることは注目に値する．一方，事前知識を使用しない手法の中では，$PPMO$ が D に次いで高い global reward 値となった．

一方，図5.6に示した結果は，文献 [42] の結果を再現できた条件で得られたものであるが，学習の設定を少し変えるだけで結果は大きく変わる．文献 [42] 及び本章で行った実験のように 1 ステップでBPD を学習する場合，Q 学習におけるパラメータである学習率 α は高くても良く，学習率を高くすると収束が早くなる．そこで，学習率を増加させた $\alpha = 0.4$ の場合についても実験を行い，その結果を図5.7に示した．

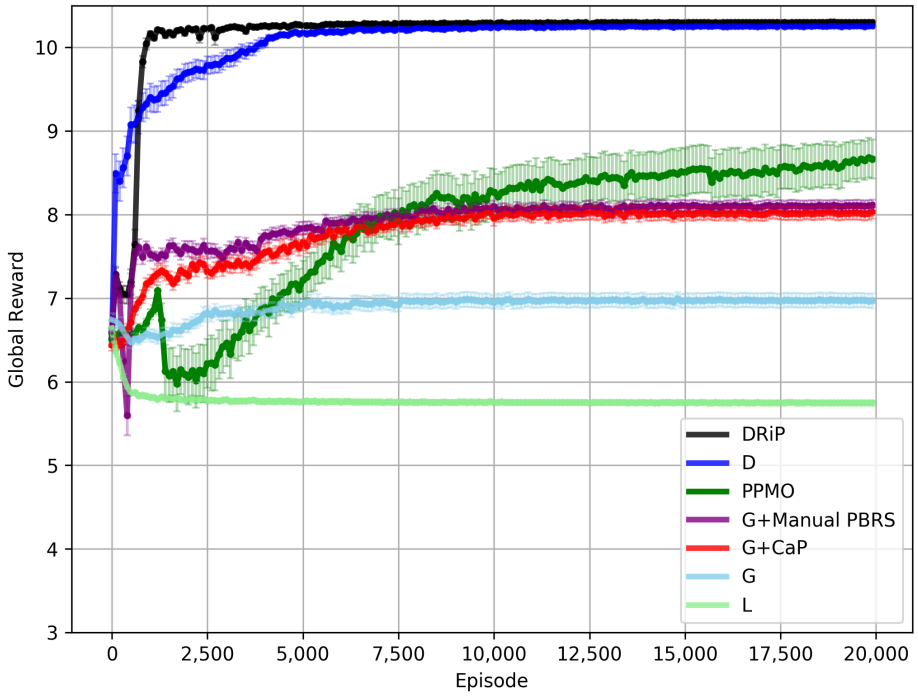


図5.6: BPD results with the default setting

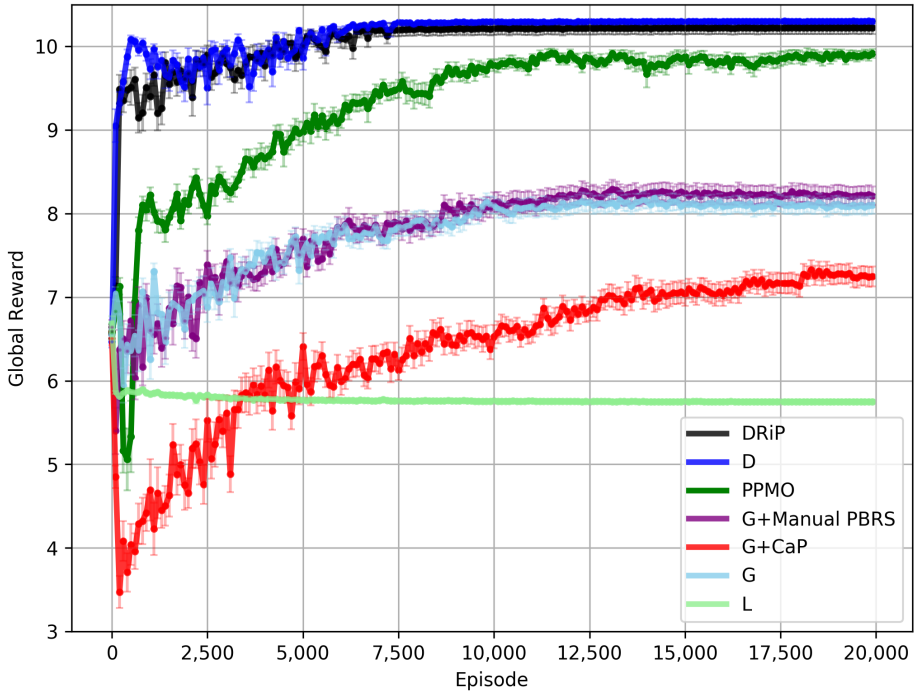


図5.7: BPD results with $\alpha = 0.4$

α を高くしたことにより G による学習が大きく改善され，それに伴って $PPMO$ の結果も改善し，一方で $G + CaP$ の結果は悪化した．この結果，この問題領域における $PPMO$ によって得られる方策の性能は，同じく事前知識を用いない D に近い値にまで改善した．

5.5 結言

本論文では，評価対象のエージェントが存在しない仮想環境を用いたマルチエージェント強化学習 (MARL) の新しい報酬設計手法を提案した．提案手法では，この $n-1$ エージェントの仮想環境を用いて，VCG と同様に，他のエージェントに対するマイナスの貢献を反映した支払いを計算する．そして現実環境で評価される報酬と，他のエージェントの評価への負の貢献を反映する支払いメカニズムに基づいて評価されるペナルティとしての報酬からなる報酬を，個々のエージェントに与える．この報酬設計手法を “Penalty based on the Payment mechanism using Minus-One environment” (PPMO) と呼ぶこととした．実験では，grid world domain (GWD) と beach problem domain (BPD) の 2 つのシナリオを用いた．その結果，どちらのシナリオにおいても，提案する PPMO は，従来手法である difference reward に近い性能を発揮することがわかった．さらに，difference reward は，評価対象のエージェントの不在を即座に評価可能という前提を必要とする一方，提案する PPMO は必要としない点が提案手法の最大の利点である．一方，PPMO の欠点はその高い計算コストにある．PPMO では，本来学習を行う対象である n エージェントの環境に加え，n 個の $n-1$ エージェントの環境について学習を行う必要があり，非常に計算コストが高い．この計算コストと，対象環境において協調を学習することの困難性とのトレードオフとなる．従って，今後の課題として以下のような項目が挙げられる．

- 一般に，強化学習は無限回の更新を前提とした場合に収束することを証明する．マルチエージェント強化学習 (MARL) では一般にナッシュ均衡解に収束することになるが，提案する報酬設計によってその収束が早まることを証明することは容易ではない．まして PPMO のように n エージェントの現実環境と n 個の $n-1$ エージェントによる仮想環境を用いた学習のダイナミクスは非常に複雑である．PPMO によって望ましい協調が促進されることの理論的証明は今後の課題である．
- PPMO による計算コストの上昇分と，対象環境の環境複雑性に由来する協調を学習することの困難性を克服するのに要する計算コストを何らかの方法で定量化し，両者のトレードオフを可能とすることは今後の課題である．
- 本実験では PPMO と併用する現実環境における報酬として global reward G または local reward L を使用したが，difference reward D や，問題領域に固有の事前知識を

反映する PBRS 手法，difference reward をポテンシャルとして使用する DRiP と組み合わせることも可能であり，その効果を確認することは今後の課題である．

第 6 章

結論

6.1 序言

本章では，提案した各手法，エージェント間通信，学習フレームワーク及び，報酬設計法について，対象とした問題環境の特徴と有効性を評価した結果を振り返るとともに， なぜ有効であるか，またそれらの課題について述べる．特に報酬設計法は，エージェントの貢献度を評価した方法と，その評価においてエージェントの不在性を仮定することの困難性，及びその緩和方法について述べる． 最後に本研究の貢献と，今後の課題について述べる．

6.2 Leader-Follower モデル

MARL による解決を困難とする要因として，環境に関する複雑性がある．3章では，そのうちのひとつとして部分観測情報環境を取り上げた．部分観測情報環境では，エージェントは観測情報として環境の真の状態ではなく不確実性のある情報に基づいて意思決定を行う必要が生じる．不確実性によって計算複雑性が著しく増大し，学習の困難性は高まる．特に，協調を狙うエージェント間で観測能力に顕著な差があるような状況の例として，本論文3章の追跡問題では3.2.2節で設定したように，Leader が常に環境全体を観測可能であり，Follower は環境に関する観測範囲に制約があるような問題設定とした．その上で，Leader から Follower に対して非常に少ない情報量の一方向通信（Leader 指示）を行えるようにし，かつその通信内容と関連する一定の強制力（Leader 強制力）を持たせるような Leader-Follower モデルを導入した．このモデルにおいて，Leader が観測能力に劣る Follower に対し適切な指示と Leader 強制力を発揮し，かつ Follower も自身の自律性において主体性と Leader 指示を活用して，チーム内の能力差や欠損を補完し合い，チーム全体としてチームワークを高め合うような学習が行えるかどうかを確認した．

その結果，提案手法と DDPG を組み合わせた場合に最も良い結果が得られた．また Leader 指示と Leader 強制力では，Leader 指示よりも Leader 強制力が結果に貢献している

ことが分かった．提案手法により学習した Predator がチームとして，他の手法により学習した Predator チームよりも統計的により多く Prey を捕捉したことは確認できたが，本節では提案手法がどのように作用したのか具体的に分析することを試みる．エージェントの獲得した行動方策により，Leader の Follower に対する指示内容と強制力の内容を定性的に観察する．図6.1〜図6.12に，提案手法（表3.3に示したケース (1)）で実施した 10 回の学習によって得られた 10 個の学習済みモデルの一例を用いた実行例 [61]を示す．図は 1 ステップ毎のエージェントごとの移動経緯を示している．

各図は, 緑が Prey, 青が Predator-Leader を示し, 赤, 紫及び灰色の丸が Predator-Follower を示す．緑の点線とそれに繋がる領域端の小点がその時点での Prey の目標位置を示す．各図下段のグラフは，横軸が Leader の action のうち Leader 指示及び Leader 強制力に繋がる要素，すなわち式3.13の $[a_i^6, a_i^7, \ldots, a_i^{12}]$ に対応し，縦軸がそれぞれの強度を示す．Leader 強制力は，式3.14と式3.14 に従って決定され，式3.16の通り各 Follower に合成される．下段グラフにおける横軸では，式3.15で定義される力の方向を図上の x,y 方向に対応させて分かりやすいように None, Left, Right, Down, Up, Come, Away と表記した．Leader 指示は式3.14の通り，$[a_i^6, a_i^7, \ldots, a_i^{12}]$ のうち最も大きい値に対応する方向の力だけが式3.15の通りに合成される．また，Leader 指示として Follower の観測情報として入力するのも式3.17のように結果として合成する力の方向である．そのため，Leader 強制力として Follower に印加される Leader 強制力の方向，そして Leader 指示の方向は同一となる．Leader 強制力及び Leader 指示の方向を各図下段のグラフ右端に None, Go Left, Go Right, Go Down, Go Up, Come On, Go Away のいずれかとして示した．

まず，図6.1の Step 1 では Prey が領域中央の右上寄りに，そして Predator がそれを囲むように位置している．Prey は左下に向かおうとしており，Leader は Follower に対し下方向に行くように指示している．この時，グラフ下段を見ると Left と Down に出力があり，Down の方が大きいため Leader 指示及び Leader 強制力としては下方向となっている．次に図6.2の Step 2 では，Prey は依然として領域左下を目標点としているが，Leader 指示は Left と Down に出力があり Left の方が値が大きくなったため，Leader 指示及び Leader 強制力は左方向に変化している．図6.3の Step 3 及び図6.4の Step 4 では，Prey の目標位置は領域左上に変化し，Leader 指示及び Leader 強制力は引き続き左方向を示している．図6.5の Step 5 では，Prey の目標位置は領域左上のままであるが，Leader 指示及び Leader 強制力は上方向となり，図6.6の Step 6 では Prey の目標位置が領域右上に変化している．続く図6.7の Step 7 及び図6.8の Step 8 では Leader 指示及び Leader 強制力の方向は右方向へ変化し，さらに図6.9の Step 9 及び図6.10の Step 10 では Prey の目標位置が領域右下に，Leader 指示及び Leader 強制力の方向も下方向に変化している．図6.11の Step 11 では Prey

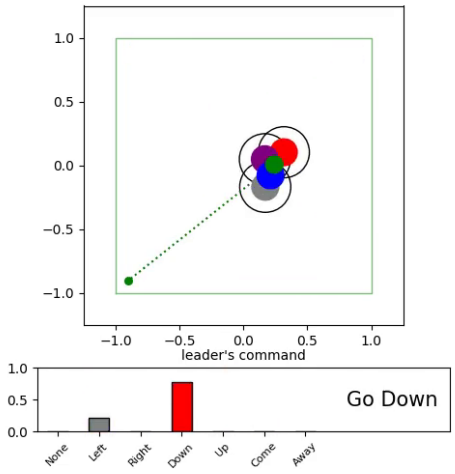


図6.1: ケース (1) 学習済みモデル実行例 Step 1

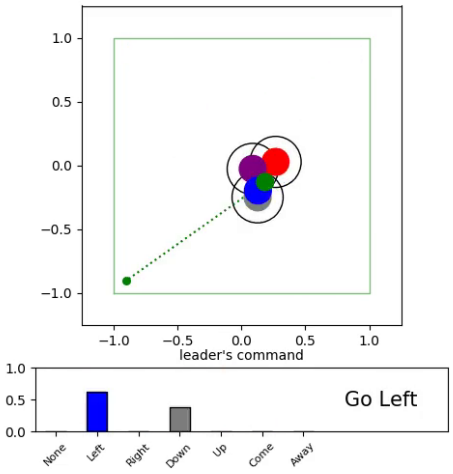


図6.2: ケース (1) 学習済みモデル実行例 step 2

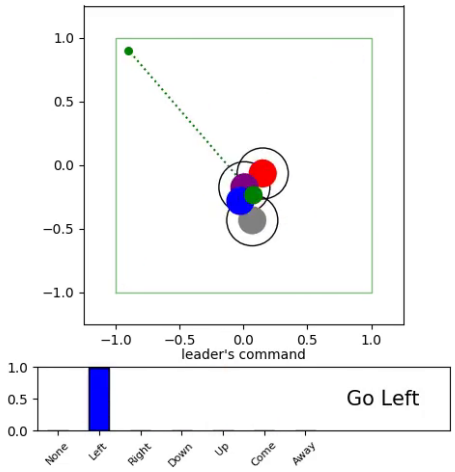


図6.3: ケース (1) 学習済みモデル実行例 step 3

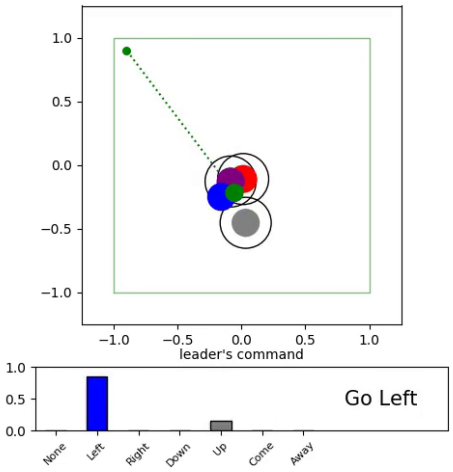


図6.4: ケース (1) 学習済みモデル実行例 step 4

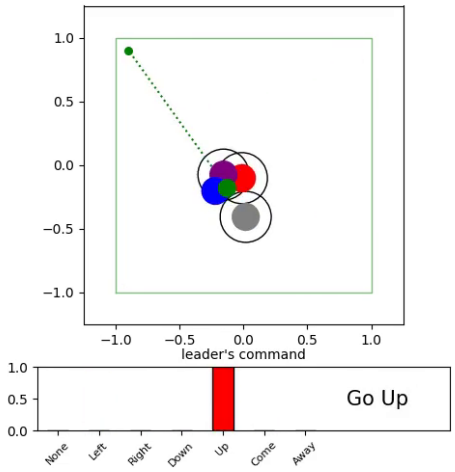


図6.5: ケース (1) 学習済みモデル実行例 step 5

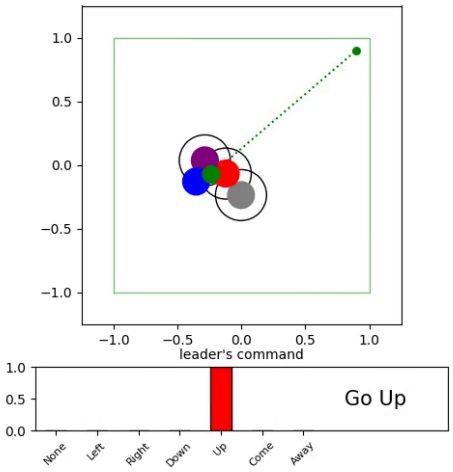


図6.6: ケース (1) 学習済みモデル実行例 step 6

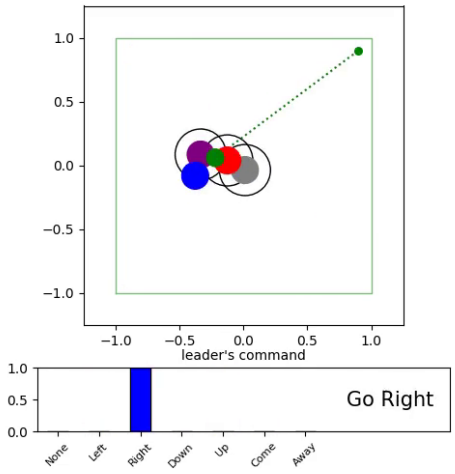


図6.7: ケース (1) 学習済みモデル実行例 step 7

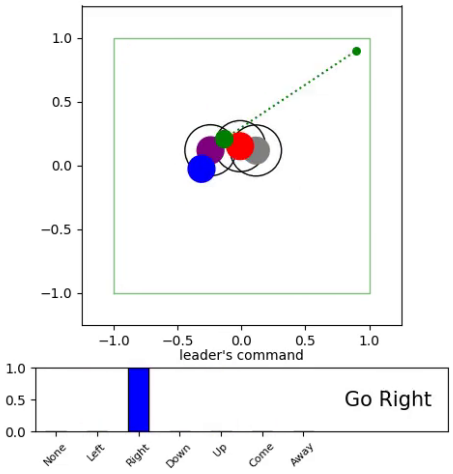


図6.8: ケース (1) 学習済みモデル実行例 step 8

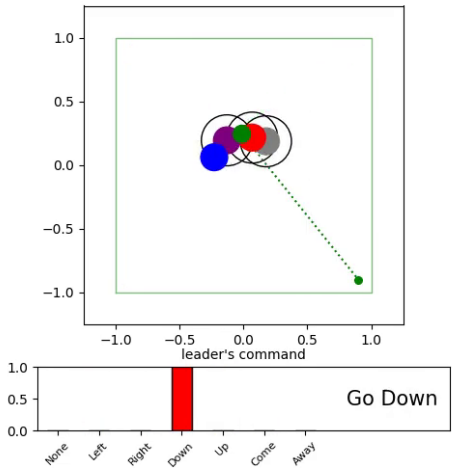


図6.9: ケース (1) 学習済みモデル実行例 step 9

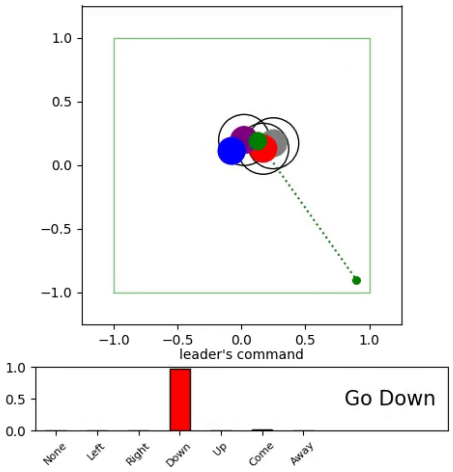


図6.10: ケース (1) 学習済みモデル実行例 step 10

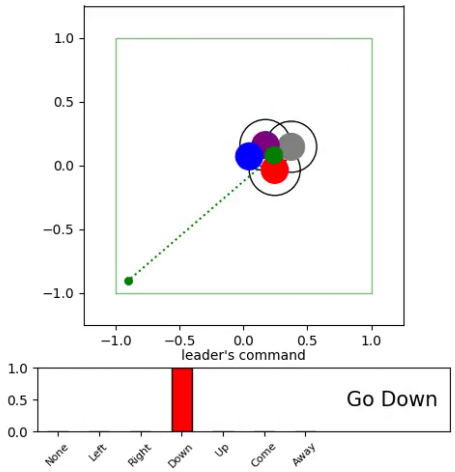


図6.11: ケース (1) 学習済みモデル実行例 step 11

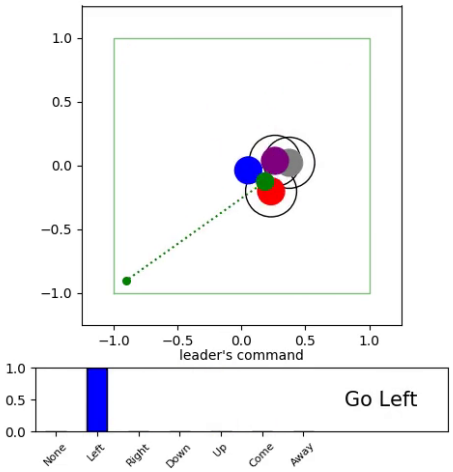


図6.12: ケース (1) 学習済みモデル実行例 step 12

の目標位置が領域左下に，Leader 指示及び Leader 強制力の方向は下方向のままであるが，図6.12の Step 12 では Leader 指示及び Leader 強制力の方向は左方向に変化している．以上から，Leader 指示は常に Prey の移動目標位置の方向を指示している．Prey の移動目標位置は Predator 全員の配置で決定するため，Prey を領域中央付近に留めるためには図6.1から図6.12の一連の経緯のように，Predator がチームとして協力して領域中央付近で移動し続ける必要がある．本環境では，これが容易ではない．例えば，各図中灰色の Follower エージェントは，図6.1の Step 1 及び図6.2の Step 2 では自身が Prey を観測可能な範囲に捉えているが，図6.3の Step 3 で Prey は観測範囲から外れている．その後，図6.10の Step 10 まで Prey を観測できていない．この間，灰色の Follower エージェントにとっては Prey の存在場所は把握することができない状況で，Leader 指示及び Leader 強制力だけを頼りに自身の移動方向を決定しなくてはならない．灰色の Folllower エージェントは，Prey を観測していない状況においても適切に移動し，再度 Prey を観測可能な範囲に捉え，捕捉にも成功している．そのためには，Follower に対して一律に出される Leader 指示及び Leader 強制力に愚直に従うだけでは不可能であり，Leader 指示及び Leader 強制力に従いつつも，自身の自律性も同時に発揮する必要がある．従って提案手法では，Leader による中央集権的な指示及び強制力と，Follower 自身の自律性の両立がチームとして達成されたことがわかる．

また Leader 指示よりも Leader 強制力が結果に貢献した理由として，本論文では3.2.2節で設定したように，Leader が常に環境全体を観測可能とし，Follower には観測範囲の制限を設定したため，Follower のみが保有する独自の情報が存在しない点が挙げられる．Follower 独自の情報が存在しないことにより Follower が自身の独自性を発揮する必要性が相対的に低下し，Leader 強制力が強い効果を持つ一方で，Follower にとって参考情報に過ぎない Leader 指示は有効ではなかったと考える．Follower しか保有しない情報が存在する環境では，Leader 強制力の重要性が低下し，Leader 指示を参考に Follower が独自性を発揮することが重要になるはずである．エージェント間での保有情報の偏り方と，それを補完することが期待されるエージェント間通信及び強制力の強さのあり方について，今後調査が必要である．

現実問題では，環境に関する全ての情報を観測し把握したうえで意思決定するようなエージェントは考えられない．従って部分観測情報環境への対処は重要である．部分観測情報環境である現実問題適用を見据えた学習エージェントによるチームワークの実現を目指すとき，エージェント間通信により集約した情報に基づき環境に関する状況把握を行って，中央集権的なチームの統率と，エージェント固有の情報を活かした個々のエージェントの自律性による分権とをうまく整合させることが必要となる．本論文で設定した問題設定と提案した Leader-Follower モデルのあり方はその一例に過ぎないが，学習エージェントによるチーム

ワークの現時点で実現可能な具体例を示せたものと考える．

一方，問題設定とそれに適した組織構成の在り方は無数に考えうる．エージェント間通信の帯域制約や遅れ，エージェント数などの問題設定に関するファクターと，その問題において望ましいチームワークを実現するための中央集権性と分権性（entralized/De-centralized）といったあるべき組織構造の間の相関関係について，今後体系的な整理が必要である．例えば，本研究では観測能力のみに差があり，その他の諸元は同一のエージェントのチームを想定し，常に環境全体を観測可能な エージェントを Leader，環境のごく一部しか観測できないエージェントを Follower とした．このように能力差の優劣が明確な場合は，固定的な Leader と Follower の役割配分でも違和感は生じにくい．一方，観測能力の差が状況によって変化する場合や，移動速度など他の能力にも差がある場合などの複雑なヘテロエージェントを想定した場合は，固定的な Leader と Follower の割り当てが適さない状況もあり得る．局面によって Leader が入れ替わったり，チームの中でさらにグループを分けて Leader-Follower を構成するなど，組織構造の取り得る形態は無数に考えられる．能力差を相互に補完し，チームとしての効用を最大化する組織構造の在り方として，人間によるチームでは考えられないような，自ら学習する自律エージェント群ならではの形態もあるかも知れない．能力差や能力差に起因する保有情報の偏在などを考慮した組織構造の整理が期待される．

6.3 学習フレームワーク

3章では，学習の初期段階において競争的環境にある一方のチームに対し，いわば勝ち方を教示するために，もう一方のチームが「あえて負ける」行動を教示するカリキュラム学習を提案した．具体的には，学習初期の 5,000 ステップまでの間，Prey が各 Predator に順番に接触していく「あえて負ける」行動によって，Predator のエージェントが Prey を捕捉することによって報酬を得る方法を教示した．また，カリキュラム学習が有効であることを実験によって示した．表3.3に示した実験ケースのうち，カリキュラムありのケース番号 (1)〜(4) 及び (6) についての学習の推移を示す図3.3〜図3.6及び図3.8では，5,000 ステップまでの間，各図上段の Mean rewards of Predators が急激に上昇しているが，これは Prey が積極的に Predator に接触することで Predator が報酬を得ていることを示している．ここでカリキュラム学習の有無による違いは，カリキュラム学習の有無以外の条件が同一であるケース番号 (1) と (5) を比較することで可能である．図3.3と図3.7を見比べると，興味深いことにカリキュラム終了から 20,000 ステップまでの間はカリキュラム学習を適用していない図3.7の方が Predator-Leader と各 Follower の Mean rewards of Predators の値の立ち上がりが早い．しかし 20,000 ステップよりも後は図3.3の方が緩やかに上昇を続ける．その理由の一つとして

考えられるのは，Follower の活躍有無である．同各図中段の Stacked mean collision counts of each Predators のグラフを比較すると，図3.3の縦軸Mean collision counts の値は Leader と各 Follower に大きな偏りなく増加を続けている一方，図3.7では Leader の Mean collision counts の値が大きく，Follower 2，Follower 3 の順で小さくなり，Follower 1 ではほとんど観測されないように，大きな偏りがある．Leader の Mean collision counts は図3.3よりもむしろ図3.7方が大きく，カリキュラムなしの場合，常に Prey の位置を観測可能な Leader が Prey を多く捕捉し，Follower はあまり Prey を捕捉できていないことが示唆される．つまり，Leader と Follower が協力し全体としての能力を発揮するチームワークとしては，提案するカリキュラム学習を適用して学習したケース (1) のほうが高いといえる．以上より，提案するカリキュラム学習は，学習によるチームワーク獲得に有効であることが分かる．

また，一般に強化学習の学習の良否は報酬値が単調増加する様子で示すことが多いが，競争環境では互いに戦略を変化させるため，図3.8や図3.9のように単調増加とはならないことが多い．このような環境における学習では，より多くの学習を行ったモデルが必ずしも優れているわけではなく，むしろ学習の途中段階で最高性能のモデルが得られていることが一般的である．本実験では，学習中は 1,000 エピソード毎にその時点の学習済みモデルを使用して 1,000 ステップ分の追跡問題を実行して Prey を捕捉した回数を集計し，学習を通じて最大回数のモデルを学習済みモデルとして保存する Train and evaluation のフレームワークを構成した．Train and evaluation 自体に新規性はないが，MADDPG のフレームワーク [1] には実装されていないことから，実験にあたり独自に実装し，実際に競争環境において有効であることが確認できた．実用的な学習結果を得るためには，こうした地道な改善方法を組み合わせることもまた大事である．

本実験では Prey 捕捉時に全ての Predator に同じ報酬を与えるいわゆる global reward を用いた．しかし，Prey の捕捉は Predator 同士の協力の結果であって，本来であれば貢献度に応じて適切に報酬を分配することが望ましい．そのため，4章以降において，エージェントごとの貢献度に応じて協力に対するインセンティブとして適切に報酬を分配するための報酬設計について検討した．

6.4 VCG の支払いに基づく報酬設計法

4章及び5章において，Vickrey-Clarke-Groves (VCG) メカニズムによる支払いアルゴリズムをベースに，社会的効用に基づくペナルティを反映した報酬をマルチエージェント強化学習に適用する，新しい報酬設計法を提案した．VCG の支払いは，対象となるエージェントが存在する場合と存在しない場合の他のエージェントによって決定される値の合計の差によって，エージェントの貢献度を測定する．この方法には，類似の従来手法である

difference reward にも共通する課題として，対象となるエージェントが存在しないことをどのように仮定するかという問題がある．4章ではエージェントの不在性を即座に評価できる問題を取り扱い，さらに5章ではより汎用性を高めるための方法について提案した．次節以降では，それぞれの結果について考察する．

6.4.1 エージェントの不在性評価が可能な場合

4章では，VCG と同様に，他のエージェントの評価に対する負の貢献度を反映した支払いを定義し，個々のエージェントには，個人の行動のみで評価される local reward と組み合わせて，支払いに基づいて評価されるペナルティとしてのネガティブな報酬を与える報酬設計法（RDPM）を提案した．この報酬設計を適用し，highway driving problem domain（HDD）と predator-prey domain（PPD）の 2 つのシナリオにおいて深層強化学習を実施する実験を行い，実験ではいずれのシナリオにおいても提案手法を用いた場合に既存研究と比較して良好な結果が得られた．

HDD では、RDPM が global reward (GR) や difference reward (DR) よりも優れた結果を示した．比較的単純な問題領域である HDD では，DR と RDPM の両方が学習を収束に導くことができたが，GR は学習が不安定となった．さらに，RDPM で学習したエージェントは，ベンチマークテストで常に高い Mean Global Reward 値を達成し， GR や DR によるエージェントとの性能差は有意であることを示した．また，HDD よりも複雑な PPD 問題領域では，RDPM を報酬に適用することで，GR や DR を適用するよりも高いスコアが得られることを実証した．

しかし，本提案手法にもいくつか課題が存在する．例えば，PPD では迷惑料を評価するための状態価値関数を Prey と Predator 間の距離の最大値を負とした値としたが，この場合，最大距離でない Predator にとっては他のエージェントに対する迷惑料が発生しない．全てのエージェントが他のエージェントに対して与えている影響を考慮することのできる状態価値関数を設計することができれば，支払い額である $p_i(\theta)$ の効果を向上することができる． 提案手法では，HDD の例のように各エージェントが別々に独自の価値関数を持つ方がより効果的である．実際，VCG メカニズムを用いたオークションなどのケースでは，オークションに参加するエージェントはそれぞれが独自の価値関数を持っている．

また，特に評価対象のエージェントの不在性を仮定する RDPM や DR では，あるエージェントが存在しなかった場合を容易に仮定して評価できる必要があり，協調問題を取り扱う場合には特に強い制約となる．協調タスクは通常，複数のエージェントが依存する部分的な作業を同時に分割して処理することが要求されるため，エージェントが存在しない場合の状態評価は単純な和や積では表現できないからである． そのため5章において，この制約を緩

和する方法を検討した．

6.4.2 適用可能性の向上

4章で提案した RDPM や DR では，一般に報酬が例えば $R(S) = R(s_i) + R(S_{-i})$ のように，各エージェントの個々の状態の和や積で容易に評価できる問題を扱っており，エージェントの不在性を即座に評価できる問題に特化している．この仮定は，エージェント間の協調を扱う際に非常に強い制約となる． そこで，5章において，実際に学習しようとしている対象環境に加えて，エージェント数が 1 つ少ない仮想的な環境を用意し，$n-1$ エージェントの環境と実際に学習しようとしている n エージェントの環境の間のエージェントの本質的な価値評価の差に基づいて報酬を計算する方法（PPMO）を提案した．実験では，grid world domain (GWD) と beach problem domain (BPD) の 2 つのシナリオを用いた．その結果，どちらのシナリオにおいても，提案する PPMO は，従来手法である difference reward に近い性能を発揮することがわかった．さらに，difference reward は，評価対象のエージェントの不在を即座に評価できるという前提を必要とする一方，提案する PPMO は必要としない．この一般的な適用性が提案する PPMO の最大の利点である．

強化学習は多数回の試行を要するため，現実問題適用を想定した場合であっても学習は現実環境ではなくシミュレーション環境を用意して行うことが多い．シミュレーション環境で学習を行うのであれば，エージェント数が 1 つ少ない仮想的な環境を作成することも可能であり，本手法を適用可能である．ただし，現実環境中で学習を行うような場合は提案手法は適用困難であるという課題がある．

また，PPMO の最も大きな欠点はその高い計算コストにある．PPMO では，本来学習を行う対象である n エージェントの環境に加え，n 個の $n-1$ エージェントの環境について学習を行う必要があり，n エージェントの環境だけで学習を行う場合に比べて，約 n 倍の計算コストがかかる．この計算コストと，対象環境において協調を学習することの困難性とのトレードオフとなる．人間が知識集約的に協調エージェントを設計する場合と，計算機を用いた学習によって協調エージェントを得ることの時間的コストと得られるエージェントの性能を定量的に比較することは容易ではないが，提案手法が一つの選択肢となりうる．

6.5 本研究の貢献

本研究では，現実環境への適用を見据えた困難性の高い環境を想定したマルチエージェント強化学習において，一つのチームとして目的に志向して行動するようなエージェントの一群を得るための提案として，主として 3 章においてエージェント間通信，学習フレームワーク及び報酬設計法について議論した．それぞれについて実験を通じて有効性を確認し，

今後の課題について述べた．4 章に示した報酬設計法では，メカニズムデザインの考え方を MARL の報酬設計に応用する方法を提案し，メカニズムデザインを現実に動作する MARL の報酬設計に反映しようとしたとき，エージェントの不在性をどのように評価するかという点に課題があることを明らかにし，5 章においてその解決方法を提示した．

現実環境は複雑であり，エージェント間の協調を組み合わせ最適化問題としてモデル化すると，容易に NP 困難以上の複雑性のある問題となる．そのため，対象問題の複雑性に関する特性と，狙うエージェントの特性に応じて何らかの緩和を行う必要がある．そのうえで，エージェントが遭遇する全ての状況をあらかじめ設計者が予想することができないような環境や，解法をプログラムとして実現することが困難な場合には，強化学習を適用することが一つの選択肢となる．その際に本論文で提案した各手法は，従来の課題であった環境複雑性による学習困難性の緩和と，学習エージェント間のインセンティブ設計を反映した報酬設計によるマルチエージェント強化学習に対する一助となる．

本研究の成果は，人間がそう遠くない将来に直面する，学習によって駆動する多数のエージェントと共生する社会，すなわち AI エージェント同士，あるいは人間同士，さらには人及び AI エージェントが混然一体となった社会状況において，互いに望ましいチームワークを実現するための中央集権性と分権性（Centralized/De-centralized）のあり方について，重要な示唆と可能性を示したと考える．

6.6 今後の課題

各章あるいは本章の各節において述べた各種提案に対する個別の課題に加え，本研究の総括としての課題として次の 2 点を挙げる．

1 点目は，チーミングのための望ましい汎用的な組織構造の解明である．本研究では，3章で部分観測環境として具体的な追跡問題の問題設定を行って Leader-Follower モデルを提案した．具体的な問題設定としたのは，暗黙的に，望ましい組織構造には対象問題に依存する部分があると考えたからである．対象問題によってソリューションが異なるのは自然であるともいえるが，一方で，全く同じ問題設定でなくとも，本研究で提案した Leader-Follower モデルが適用可能で効果的な他の問題もまた，存在するはずである．適用範囲や適用効果の限界をより明確化するためには，提案の抽象化と汎化が必要であると考えており，さらなる理論的深堀が期待される．

2 点目は，メカニズムデザインを応用した報酬設計について，4章で述べた RDPM と5章で取り上げた PPMO という 2 つの手法ともに，確かにその報酬設計によって望ましい学習結果が導かれるということを実験的に示した一方，数学的証明を欠いている点である．例えば図5.1を一見して分かるように，高度に抽象的かつ理論的な VCG メカニズムを報酬設計

に適用する MARL の学習ダイナミクスは非常に複雑であり，エージェントの価値が逐次的に更新されていく MARL において報酬設計が MARL の方式によらず必ず良い結果を導くかどうかを数学的に記述・評価するのは難しい．一般に，強化学習において学習方式の良性を述べる際には，無限回の更新，すなわち状態と行動のすべての空間を探索し尽くした場合に価値や方策が収束することで示すなど，何らかの前提を置いて議論されるのが普通である．例えば，Nash Q-learning [71] でも，NashQ 関数が縮約演算子であり，無限回の更新を前提として鞍点またはナッシュ均衡に向かて収れんすることと，ナッシュ均衡解の時に pay off が最大となることを証明している．まして深層強化学習を用いた場合の学習ダイナミクスの数学的記述は非常に難解である．学習の過程において発生する一時的な学習エージェント同士の関係性の時系列的な変化まで考慮した記述・評価を行うには，無限回の更新を前提とした証明だけでは不十分である．無限回の更新といった前提の証明が機械学習の実用上の効果に対してどの程度意味があるのかどうかも含めて，数学的証明は今後の課題とする．

MARL の現実適用では，対象問題を適切にモデル化し，Q 学習や深層学習をはじめ多数の学習方式の中から，対象とする問題に応じて適する方法を選択し，問題領域に固有の事前知識を活用することも考慮して報酬を設計して学習を実施し，評価することが必要になる．より良い結果を導くために，モデル化，学習方式の選択，報酬の設計，及び評価の各段階においてありとあらゆる工夫をすることが必要であり，本論文の各提案手法が活用可能である．

参考文献

[1] R. Lowe *et al.*, “Multi-agent actor-critic for mixed cooperative-competitive environments,” *Neural Information Processing Systems (NIPS)*, 2017.

[2] S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, 3rd ed. Prentice Hall Press, 2009.

[3] L. Buşoniu *et al.*, “A comprehensive survey of multi-agent reinforcement learning,” *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, vol. 38, no. 2, pp. 156–172, Mar. 2008.

[4] V. Mnih *et al.*, “Playing atari with deep reinforcement learning,” *CoRR*, vol. abs/1312.5602, 2013. [Online]. Available: http://arxiv.org/abs/1312.5602

[5] V. Mnih *et al.*, “Human-level control through deep reinforcement learning,” *Nature*, vol. 518, no. 7540, pp. 529–533, Feb. 2015. [Online]. Available: http://dx.doi.org/10.1038/nature14236

[6] G. Brockman *et al.*, “Openai gym,” *CoRR*, vol. abs/1606.01540, 2016. [Online]. Available: http://arxiv.org/abs/1606.01540

[7] L. Panait and S. Luke, “Cooperative multi-agent learning: The state of the art,” *Autonomous Agents and Multi-Agent Systems*, vol. 11, no. 3, pp. 387–434, Nov. 2005. [Online]. Available: http://dx.doi.org/10.1007/s10458-005-2631-2

[8] P. Hernandez-Leal, M. Kaisers, T. Baarslag, and E. M. de Cote, “A survey of learning in multiagent environments: Dealing with non-stationarity,” *CoRR*, vol. abs/1707.09183, 2017.

[9] T. Groves *et al.*, “Incentives in teams,” *Econometrica*, vol. 41, no. 4, pp. 617–631, 1973.

[10] L. Bu, R. Babu, B. De Schutter *et al.*, “A comprehensive survey of multiagent reinforcement learning,” *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 38, no. 2, pp. 156–172, 2008.

[11] K. Zhang, Z. Yang, and T. Basar, “Multi-agent reinforcement learning: A selective

overview of theories and algorithms," *CoRR*, vol. abs/1911.10635, 2019. [Online]. Available: http://arxiv.org/abs/1911.10635

[12] *The Cambridge Academic Content Dictionary.* Cambridge University Press, 2021.

[13] M. Tambe, "Towards flexible teamwork," *Journal of Artificial Intelligence Research*, vol. 7, pp. 83–124, 1997.

[14] 野田五十樹, "ロボカップシミュレーションリーグ," in ゲームプログラミングワークショップ *2002* 論文集, vol. 2002, no. 17, nov 2002, pp. 22–27.

[15] W. M. Muir and H. W. Cheng, "Chapter 9 - genetic influences on the behavior of chickens associated with welfare and productivity," in *Genetics and the Behavior of Domestic Animals (Second Edition)*, second edition ed., T. Grandin and M. J. Deesing, Eds. Academic Press, 2014, pp. 317–359. [Online]. Available: https://www.sciencedirect.com/science/article/pii/B9780123945860000093

[16] M. Heffernan, "Forget the pecking order at work," 2015, https://www.ted.com/talks/margaret_heffernan_why_it_s_time_to_forget_the_pecking_order_at_work?language=en.

[17] *The Cambridge Business English Dictionary.* Cambridge University Press, 2021.

[18] R. Nair and M. Tambe, "Hybrid bdi-pomdp framework for multiagent teaming," *Journal of AI Research (JAIR)*, vol. 23, pp. 367–420, 2005.

[19] R. S. Sutton and A. G. Barto, "Reinforcement learning: An introduction," *IEEE Transactions on Neural Networks*, vol. 16, pp. 285–286, 1998.

[20] T. P. Lillicrap *et al.*, "Continuous control with deep reinforcement learning," *ArXiv e-prints*, Sep. 2015.

[21] M. Hessel *et al.*, "Rainbow: Combining improvements in deep reinforcement learning," *CoRR*, vol. abs/1710.02298, 2017. [Online]. Available: http://arxiv.org/abs/1710.02298

[22] C. Berner, G. Brockman, B. Chan, V. Cheung, P. Debiak, C. Dennison, D. Farhi, Q. Fischer, S. Hashme, C. Hesse, R. Józefowicz, S. Gray, C. Olsson, J. Pachocki, M. Petrov, H. P. de Oliveira Pinto, J. Raiman, T. Salimans, J. Schlatter, J. Schneider, S. Sidor, I. Sutskever, J. Tang, F. Wolski, and S. Zhang, "Dota 2 with large scale deep reinforcement learning," *CoRR*, vol. abs/1912.06680, 2019. [Online]. Available: http://arxiv.org/abs/1912.06680

[23] O. Vinyals, I. Babuschkin, W. Czarnecki, M. Mathieu, A. Dudzik, J. Chung, D. Choi, R. Powell, T. Ewalds, P. Georgiev, J. Oh, D. Horgan, M. Kroiss, I. Danihelka,

A. Huang, L. Sifre, T. Cai, J. Agapiou, M. Jaderberg, and D. Silver, "Grandmaster level in starcraft ii using multi-agent reinforcement learning," *Nature*, vol. 575, 11 2019.

[24] Y. Huang, S. Wu, Z. Mu, X. Long, S. Chu, and G. Zhao, "A multi-agent reinforcement learning method for swarm robots in space collaborative exploration," in *2020 6th International Conference on Control, Automation and Robotics (ICCAR)*, 2020, pp. 139–144.

[25] J. Kober, J. Bagnell, and J. Peters, "Reinforcement learning in robotics: A survey," *The International Journal of Robotics Research*, vol. 32, pp. 1238–1274, 09 2013.

[26] S. Shalev-Shwartz, S. Shammah, and A. Shashua, "Safe, multi-agent, reinforcement learning for autonomous driving," *CoRR*, vol. abs/1610.03295, 2016. [Online]. Available: http://arxiv.org/abs/1610.03295

[27] C. Yu, X. Wang, X. Xu, M. Zhang, H. Ge, J. Ren, L. Sun, B. Chen, and G. Tan, "Distributed multiagent coordinated learning for autonomous driving in highways based on dynamic coordination graphs," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 2, pp. 735–748, 2020.

[28] H. Mao, Z. Gong, and Z. Xiao, "Reward design in cooperative multi-agent reinforcement learning for packet routing," 2018. [Online]. Available: https://openreview.net/forum?id=r15kjpHa-

[29] C. Undeger and F. Polat, "Multi-agent real-time pursuit," *Autonomous Agents and Multi-Agent Systems*, vol. 21, pp. 69–107, 2009.

[30] R. E. Wang, M. Everett, and J. P. How, "R-MADDPG for partially observable environments and limited communication," *CoRR*, vol. abs/2002.06684, 2020. [Online]. Available: https://arxiv.org/abs/2002.06684

[31] W. S. Lovejoy, "Computationally feasible bounds for partially observed markov decision processes," *Operations Research*, vol. 39, no. 1, pp. 162–175, 1991. [Online]. Available: https://doi.org/10.1287/opre.39.1.162

[32] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, pp. 1735–80, 12 1997.

[33] M. J. Hausknecht and P. Stone, "Deep recurrent q-learning for partially observable mdps," *CoRR*, vol. abs/1507.06527, 2015. [Online]. Available: http://arxiv.org/abs/1507.06527

[34] Y. J. Park, Y. S. Cho, and S. B. Kim, "Multi-agent reinforcement learning with

approximate model learning for competitive games," *PLOS ONE*, vol. 14, no. 9, pp. 1–20, 09 2019. [Online]. Available: https://doi.org/10.1371/journal.pone.0222215

[35] J. Jiang and Z. Lu, "Learning attentional communication for multi-agent cooperation," in *International Conference on Neural Information Processing Systems*, 2018, pp. 7265–7275.

[36] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, ser. NIPS'14. MIT Press, 2014, pp. 2672–2680. [Online]. Available: http://dl.acm.org/citation.cfm?id=2969033.2969125

[37] D. H. Wolpert and K. Tumer, "An introduction to collective intelligence," *CoRR*, vol. cs.LG/9908014, 1999. [Online]. Available: https://arxiv.org/abs/cs/9908014

[38] A. Agogino and K. Tumer, "Unifying temporal and structural credit assignment problems," in *Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems - Volume 2*, ser. AAMAS '04. IEEE Computer Society, 2004, pp. 980–987. [Online]. Available: http://dl.acm.org/citation.cfm?id=1018410.1018852

[39] A. K. Agogino and K. Tumer, "Analyzing and visualizing multiagent rewards in dynamic and stochastic environments," *Journal of Autonomous Agents and Multiagent Systems*, pp. 320–338, 2008.

[40] A. Agogino and K. Tumer, "Multi-agent reward analysis for learning in noisy domains," in *Proceedings of the Fourth International Joint Conference on Autonomous Agents and Multiagent Systems*, ser. AAMAS ' 05. Association for Computing Machinery, 2005, pp. 81–88. [Online]. Available: https://doi.org/10.1145/1082473.1082486

[41] A. Y. Ng, D. Harada, and S. Russell, "Policy invariance under reward transformations: Theory and application to reward shaping," in *In Proceedings of the Sixteenth International Conference on Machine Learning*. Morgan Kaufmann, 1999, pp. 278–287.

[42] S. Devlin, L. Yliniemi, D. Kudenko, and K. Tumer, "Potential-based difference rewards for multiagent reinforcement learning," in *Proceedings of the 2014 International Conference on Autonomous Agents and Multi-agent Systems*, ser. AAMAS '14. International Foundation for Autonomous Agents

and Multiagent Systems, 2014, pp. 165–172. [Online]. Available: http://dl.acm.org/citation.cfm?id=2615731.2615761

[43] C. HolmesParker, M. Taylor, A. Agogino, and K. Tumer, "Cleaning the reward: Counterfactual actions remove exploratory action noise in multiagent learning (extended abstract)," in *Proceedings of the Thirteenth International Joint Conference on Autonomous Agents and Multiagent Systems*, May 2014.

[44] J. M. Vidal and J. M. Vidal, "Fundamentals of multiagent systems with netlogo examples," p. 130, 2006.

[45] T. Roughgarden, "Algorithmic game theory," *Communications of the ACM*, vol. 53, no. 7, pp. 78–86, 2010.

[46] 伊藤孝行, ‘‘計算論的メカニズムデザイン,” コンピュータソフトウェア, vol. 25, no. 4, 2008.

[47] W. Vickrey, "Counterspeculation, Auctions, And Competitive Sealed Tenders," *Journal of Finance*, vol. 16, no. 1, pp. 8–37, March 1961. [Online]. Available: https://ideas.repec.org/a/bla/jfinan/v16y1961i1p8-37.html

[48] E. H. Clarke, "Multipart pricing of public goods," *Public Choice*, vol. 11, no. 1, pp. 17–33, Sep 1971. [Online]. Available: https://doi.org/10.1007/BF01726210

[49] T. Groves, "Incentives in teams," *Econometrica*, vol. 41, no. 4, pp. 617–31, 1973.

[50] H. Smets, "Le principe de la compensation réciproque: un instrument économique pour la solution de certains problèmes de pollution transfrontière," *OCDE, Direction de l' Environnement*, 1973.

[51] M. Natsuki, O. Shun, and I. Takayuki, "Reward design for multi-agent reinforcement learning with a penalty based on the payment mechanism," *Transactions of the Japanese Society for Artificial Intelligence*, vol. 36, no. 5, 09 2021.

[52] 大沢英一 *et al.*, ‘‘分散人工知能における標準的小問題,” コンピュータソフトウェア, vol. 10, no. 3, pp. 195–211, may 1993.

[53] E. OSAWA, "A metalevel coordination strategy for reactive cooperative planning," *ICMAS'95, San Francisco, USA*, pp. 297–303, 1995. [Online]. Available: https://ci.nii.ac.jp/naid/10027987962/

[54] OpenAI. (2018) Multi-agent particle environment. [Online]. Available: https://github.com/openai/multiagent-particle-envs

[55] S. L. Barton, E. G. Zaroukian, D. E. Asher, and N. R. Waytowich, "Evaluating the coordination of agents in multi-agent reinforcement learning," in *Intelligent Human*

Systems Integration 2019, 02 2019, pp. 765–770.

[56] S. Li, Y. Wu, X. Cui, H. Dong, F. Fang, and S. Russell, "Robust multi-agent reinforcement learning via minimax deep deterministic policy gradient," in *AAAI Conference on Artificial Intelligence*, vol. 33, 07 2019, pp. 4213–4220.

[57] H. Mao, Z. Zhang, Z. Xiao, and Z. Gong, "Modelling the dynamic joint policy of teammates with attention multi-agent DDPG," *CoRR*, vol. abs/1811.07029, 2018. [Online]. Available: http://arxiv.org/abs/1811.07029

[58] X. Hao, W. Wang, J. Hao, and Y. Yang, "Independent generative adversarial self-imitation learning in cooperative multiagent systems," in *International Conference on Autonomous Agents and MultiAgent Systems*, 2019, pp. 1315–1323.

[59] M. Benda, V. J. Jagannathan, and R. T. Dodhiawala, "On optimal cooperation of knowledge sources-an empirical investigation," in *Technical Report*. Boeing Advanced Technology Center, 1986.

[60] Y. Bengio, J. Louradour, R. Collobert, and J. Weston, "Curriculum learning," in *Proceedings of the 26th Annual International Conference on Machine Learning*, ser. ICML '09. ACM, 2009, pp. 41–48. [Online]. Available: http://doi.acm.org/10.1145/1553374.1553380

[61] N. Matsunami. (2021) ケース (1) と (7) で得られたモデルの行動傾向例（動画）, https://www.youtube.com/watch?v=go6pyks1krq. [Online]. Available: https://www.youtube.com/watch?v=Go6pYKS1KRQ

[62] D. Mguni, "Efficient reinforcement dynamic mechanism design," in *AAMAS Workshops*, 2019.

[63] K. Ma and P. R. Kumar, "The strategic LQG system: A dynamic stochastic VCG framework for optimal coordination," in *57th IEEE Conference on Decision and Control, CDC 2018, Miami, FL, USA, December 17-19, 2018*. IEEE, 2018, pp. 5777–5782. [Online]. Available: https://doi.org/10.1109/CDC.2018.8619894

[64] B. Edelman, M. Ostrovsky, and M. Schwarz, "Internet advertising and the generalized second-price auction: Selling billions of dollars worth of keywords," *The American Economic Review*, vol. 97, pp. 242–259, 03 2007.

[65] M. Benda, "On optimal cooperation of knowledge source," *Technical Report*, 1985.

[66] E. Osawa, "A metalevel coordination strategy for reactive cooperative planning." in *ICMAS*, vol. 95, 1995, pp. 297–303.

[67] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann ma-

chines," in *Proceedings of the 27th international conference on machine learning (ICML-10)*, 2010, pp. 807–814.

[68] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2017.

[69] P. J. Huber, "Robust estimation of a location parameter," *Ann. Math. Statist.*, vol. 35, no. 1, pp. 73–101, 03 1964. [Online]. Available: https://doi.org/10.1214/aoms/1177703732

[70] B. L. Welch, "The generalization of 'student's' problem when several different population variances are involved," *Biometrika*, vol. 34, no. 1/2, pp. 28–35, 1947. [Online]. Available: http://www.jstor.org/stable/2332510

[71] J. Hu and M. P. Wellman, "Nash q-learning for general-sum stochastic games," *The Journal of Machine Learning Research*, vol. 4, no. null, pp. 1039–1069, Dec. 2003.

[72] S. Proper and K. Tumer, "Coordinating actions in congestion problems: Impact of top-down and bottom-up utilities," *Autonomous Agents and MultiAgent Systems*, vol. 27, no. 3, pp. 419–443, 2013.

謝辞

この学位論文の執筆及び執筆に至るまでの過程において，数多くの方々のご指導とご支援を頂きました．心より感謝申し上げます．名古屋工業大学工学研究科情報工学専攻博士後期課程に入学して以来，弛まぬご指導を賜りました京都大学大学院情報学研究科社会情報学専攻の伊藤孝行教授に謹んで感謝の意を表します．伊藤孝行教授との出会いは古く，私が現在も在職している会社の業務における技術的な課題について，会社内のクローズドな議論や検討だけでは打破できないと考え，2012 年に名古屋工業大学の科学技術相談窓口を通じてご面談頂いたことに始まります．その後，様々な形でご相談を重ねご指導を賜る中で，業務上の必要性を超えて，個人として「社会的に望ましい意思決定の在り方とはどういうものか」を追求したくなり，社会人学生として伊藤孝行研究室に飛び込むことといたしました．伊藤孝行教授には，そんな私を快く受け入れて下さり，なかなか成果を出せない私を忍耐強く，親切丁寧なご指導を賜りました．ややもすると時間的制約から安易な方向に向かいがちな私の姿勢を，常に世界レベルの高い目標に向かうように，かつその目標点が達成可能であることを示唆して頂きました．そのため，高い目標に向かう過程で，本論文には書き尽くせない数多くの無形の学びを得ることができました．先生への感謝は，言葉に尽くせません．心より感謝申し上げます．

京都大学大学院情報学研究科の奥原俊特定助教には，悩み苦しむ私に積極的に手を差し伸べて頂き，誠に感謝いたします．先生の先回りするようなご支援とご配慮には，何度助けられたことか分かりません．元々研究室になかなか訪れる機会がなく他学生と情報交換する機会に恵まれない中，コロナ禍でその機会もほとんど失われ，研究の進め方や手続きに不備が生起しがちな私に暖かいご助言とサポートを頂きました．心より謝意を申し上げます．

加藤昇平教授には，伊藤先生の転出以降，快く研究室に受け入れて下さり，継続的に有益なご指摘と心温まる励ましを頂き，厚く御礼申し上げます．

京都大学大学院情報学研究科の Rafik Hadfi 特任助教には，研究論文の論理構成に対して親身になって貴重なご助言を賜り，心より感謝いたします．

当時の名古屋工業大学の研究室に所属されていた博士後期課程の在校生，卒業生らには，共に学び，研究で切磋琢磨することができました．皆様に感謝いたします．特に博士前期課

程の丹田尋氏，兵藤佑輝氏とは，課題抽出や解決方法を共に検討し，研究を推進できたことが何事にも代えがたい貴重な経験となりました．厚くお礼申し上げます．

在学中は，多くの教職員の方々，学生，友人，知人にご支援頂きました．また会社員として日々の業務の傍ら，大学で学ぶことに理解を示してくれた会社の上司，同僚，部下からも陰に陽に多大なる支援を頂きました．ここに記載しきれなかった多くの方々に感謝いたします．

また，生涯を通じて学ぶことの大切さと喜びを教えてくれ，病床で私が博士後期課程に進むと報告すると喜びと不安の入り混じった表情を見せた亡母松波廸子，常に挑戦することを背中で示した亡父松波晧介，そして今も常に学び未知なる世界に挑み続けて道を切り開き，我が人生の道標たる兄松波晴人に感謝いたします．そして上天より変わらぬ暖かい笑顔と励ましをくれた亡兄松波純也に感謝いたします．

私が業務多忙のさなか，さらに学問の道にも進むことで大変な苦労をかけたにも関わらず，支え続けてくれた妻松波由起子に感謝いたします．遊ぶ時間が制約される不満を飲み込んで協力してくれた息子の松波理央にも感謝いたします．

最後に，重ねて伊藤孝行先生と奥原俊先生，伊藤孝行研究室で共に過ごした皆さまや，家族に感謝を申し上げます．

2022 年 1 月

松波夏樹

本論文に関する研究業績

学術論文

(1). 松波夏樹, 奥原俊, 伊藤孝行, 連続空間の追跡問題における Leader 指示と Leader 強制力に基づくチームワークの学習, 人工知能学会論文誌, Vol.36, No.5, p. E-K62_1-10, 2021.

(2). Natsuki Matsunami，Shun Okuhara, Takayuki Ito, Reward Design for Multi-Agent Reinforcement Learning with a Penalty Based on the Payment Mechanism，人工知能学会論文誌　論文特集「エージェント技術とその応用 2021」, Vol.36, No.5, p. AG21-H_1-11, 2021.

査読付き国際会議

(1). Natsuki Matsunami, Shun Okuhara, Takayuki Ito, Agents that Learn to Vote for a Joint Action Through Multi-Agent Reinforcement Learning, 8th International Conference on Smart Computing and Artificial Intelligence (SCAI 2020), 9th International Congress on Advanced Applied Informatics, 2020.

国内口頭発表

(1). 松波夏樹, 丹田尋, 伊藤孝行, マルチエージェント強化学習における報酬設計へのクラーク税の導入による協調行動創発の促進, 合同エージェントワークショップ＆シンポジウム 2019 (JAWS2019), 2019.

(2). 松波夏樹, 丹田尋, 伊藤孝行, 部分観測情報に基づくマルチエージェント深層強化学習における協調戦略の獲得, 合同エージェントワークショップ＆シンポジウム 2018 (JAWS2018), 2018.

その他の研究業績

ジャーナル論文

(1). 松波夏樹, 唐鎌聡太郎, 人と AI 群が協働するチーム対戦の取組み　人間 30 人 vs 1 人のしっぽ鬼，人工知能学会論文誌, Vol.36, No.5, p. G-L45_1-6, 2021.

(2). Sotaro Karakama, Natsuki Matsunami, Masayuki Ito, Task Decomposition and Role Sharing for Real-time Human-AI Swarm Collaboration, International Journal of Smart Computing and Artificial Intelligence (IJSCAI), 2021.

査読付き国際会議

(1). Natsuki Matsunami, Sotaro Karakama, Masayuki Ito, Architecture and Interface for Collaborating with a Group of Agents in an Adversarial Game, 8th International Conference on Smart Computing and Artificial Intelligence (SCAI), 2020.