JAIST Repository
https://dspace.jaist.ac.jp/

Title	少数の記録からプレイヤの価値観を機械学習するチームプレイAIの構成
Author(s)	和田，尭之
Citation	
Issue Date	2015-03
Type	Thesis or Dissertation
Text version	author
URL	http://hdl.handle.net/10119/12785
Rights	
Description	Supervisor:池田　心，情報科学研究科，修士

JAIST
JAPAN
ADVANCED INSTITUTE OF
SCIENCE AND TECHNOLOGY

Japan Advanced Institute of Science and Technology

修 士 論 文

少数の記録からプレイヤの価値観を機械学習するチームプレイ AI の構成

北陸先端科学技術大学院大学
情報科学研究科情報科学専攻

和田 尭之

2015 年 3 月

修 士 論 文

少数の記録からプレイヤの価値観を機械学習するチームプレイ AI の構成

指導教員　池田 心 准教授

審査委員主査　池田 心 准教授
審査委員　飯田 弘之 教授
審査委員　東条 敏 教授

北陸先端科学技術大学院大学
情報科学研究科情報科学専攻

1310082 和田 尭之

提出年月: 2015 年 2 月

Copyright © 2015 by Wada Takayuki

概 要

ゲーム情報学の目標の一つに「人間を楽しませるコンピュータプレイヤ（本稿ではゲーム AI と呼ぶ)」を作ることがあげられる．その中でもこれまでのゲーム AI の研究は人間プレイヤの敵として相手を楽しませるために強い AI プレイヤの作成を目的とした研究といえる．既にオセロやチェス，将棋などといったボードゲームにおいて，ゲーム AI は人間プレイヤのプロレベルの強さに達している．さらにゲーム AI の研究の対象は現在一般的になってきているコンピュータゲームにまで広がっている．コンピュータゲームで対象となっているジャンルには一人称視点銃撃ゲーム（FPS）やリアルタイムストラテジー（RTS）などがあり，強さや動きの自然さを追求した敵のゲーム AI 作る方法が提案されてきた．しかし一方で，人間プレイヤの仲間として人間プレイヤを楽しませる研究は少ない．

市販のコンピュータゲーム特に RPG と呼ばれるジャンルでは，ゲーム AI が操作するキャラクタとチームを組んで遊べるものも多い．これらのゲームでは敵を賢くする以上に仲間のゲーム AI を賢くすることが人間プレイヤを楽しませることにつながる．チームプレイの醍醐味はコンビネーションであり，チームプレイをするゲーム AI には仲間と連携して行動することが求められる．しかし，しばしば仲間 AI プレイヤは期待に反する行動を取り，プレイヤの不満に繋がる．これはこの種のゲームに“勝つ”以外に「キャラクタをできるだけ傷つけない」「できるだけ早く勝利する」などの副目的が複数あり，AI プレイヤは人間プレイヤの“どの副目的をどの程度重視しているか”といった価値観を理解せずに行動していることが原因の一つである．副目的を共有できないため人間プレイヤの動きと連携して行動することができない．

本研究では，人間プレイヤが選択した行動から人間プレイヤの重視する副目的を推定し，それを AI プレイヤの行動選択に活用することでその人間プレイヤにとって満足度が高い AI プレイヤを生成することを目指した．まず初めに研究用途に適したルールを設定し，それに基づいたゲームを作成した．本研究では RPG の中でも現在日本での人気が高いコマンド形式 RPG を想定した，多人数順次着手確定ゲームとした．このゲームにおいて敵味方の置かれた状況は非対称であり，多くの場合プレイヤチーム側が勝利できるようになっている．しかし勝利したとしても「チームのメンバーが戦闘不能になっている」「精神力を使い果たした」などその最終状態には好ましい場合と好ましくない場合があり，しかもそれが点数などでは明示されない．このような特徴はコマンド形式 RPG において珍しいものではなく，本研究もこれを踏襲する．

本研究では,「人間は何らかの価値関数を持っており，それが一番高くなる行動を決定している」と仮定した．この仮定に基づき，人間プレイヤの選択した行動と行動を選択した時の状態を引数とする価値関数を学習する関数モデルを作成し，そのモデルが持つ可変パラメータを調整することで人間プレイヤのもつ価値関数を表現した．さらに学習した価値関数により状態を入力として行動を出力するモデルを作成し，人間プレイヤの価値関数で最善となる行動を選択することで人間プレイヤに迎合することを目指した．

提案手法では価値関数の推定のためにモンテカルロ法によるシミュレーションで平均的帰結を求めている．モンテカルロ法では“ありそうな行動”を高い確率で選択するように偏らせたシミュレーションのほうがよいことが多いことが知られているが，副目的により“ありそうな行動”は大きく変化するため，適切な固定の戦略を定めることは困難である．そこで本研究では複数の戦略を用意し，それぞれの行動それぞれの戦略ごとに平均的帰結を求めるアプローチをとった．これによりモンテカルロ法を使用したAIは戦略がない場合と比べより正確なシミュレーションを行えることで強くなり，平均的帰結の推定もより正確に行えることが示された．

評価実験では，様々な価値観を持つ仮想人間プレイヤを人工的に構成し，提案手法を適用して価値観を推定した．全く同じ価値観に基づいて行動を選択した場合の行動一致率（例えば70.6%）に対し，推定した価値観に基づいて行動を選択した場合の行動一致率（例えば67.1%）は，最悪の場合でも3.5%しか劣っていない結果を得ることができた．

さらに被験者実験を行い，実際の人間プレイヤに対してどの程度学習ができるのかを確認した．実験では，まず人間プレイヤに「ＭＰを節約したうえで勝ってください」あるいは「できるだけ早く勝ってください」といった指示を与えたうえで，味方チームの全キャラクタを操作してもらった．この戦闘を4回繰り返し，その際の行動から提案手法は“その”人間プレイヤの価値観を学習した．続いて評価フェイズに移り，コンピュータが1つのキャラクタ，人間プレイヤがもう1つのキャラクタを操作してチームを組み，同じ課題に取り組んでもらった．このときのコンピュータは，提案手法の場合と，固定の価値関数を持つモンテカルロAIの場合があり，それらをプレイヤには伏せた状態で「このAIの動きの自然さを5段階で評価してください」という質問を行った．評価結果は，どのような指示を与えた場合も，固定の価値関数のものに比べて提案手法が高く（平均で0.7点高い），提案手法が人間プレイヤの価値関数を推定してそれに迎合する行動が取れたことが示せた．

目 次

第 1 章　はじめに **1**

第 2 章　関連研究 **2**

第 3 章　着眼点とアプローチ **4**

第 4 章　提案手法 **6**
4.1　プレイヤ行動の記録 6
4.2　平均的帰結のシミュレーション 7
4.3　価値関数の推定 9
4.4　人間プレイヤが満足する行動を決定 10

第 5 章　本研究で扱うゲームの概要 **11**
5.1　キャラクタの持つパラメータ 11
5.2　行動とその効果 12
5.3　状態遷移 13

第 6 章　設定と戦略 **14**
6.1　戦闘参加キャラクタの設定 14
6.2　取りうる戦略の設定 15
6.3　予備実験 1 ：戦略の価値 16
6.4　典型的な行動の分岐点における評価値の違い 17
6.4.1　HP が少ない時 HP 優先であるのに単体攻撃を行う例 18
6.4.2　Turn 優先であるのに中回復を行う例 19
6.4.3　MP 優先であるのにグループ攻撃を行う例 20

第 7 章　シミュレーション結果の特徴量化 **21**
7.1　状態評価関数 21
7.2　重みベクトル空間 22
7.3　予備実験 2 ：行動一致率の分布 23

第 8 章　実験 **25**
8.1　人工プレイヤに対する学習実験 25

8.2 被験者実験 29

第 9 章 発展的なペナルティ加算法の提案 **30**
9.1 学習を高速化するペナルティの与え方 30
9.2 Welch 検定 31
9.3 01 ペナルティとの比較 32

第 10 章 まとめ **34**

付 録 A 実験に使用した設定 **36**

第1章　はじめに

ゲーム情報学の目標の一つに「人間を楽しませるコンピュータプレイヤ（本稿ではゲーム AI と呼ぶ)」を作ることがあげられる．近年までのゲーム AI の研究の多くは強いゲーム AI の作成を目的とし，既にオセロやチェス，将棋といったボードゲームにおいて，ゲーム AI は人間のプロレベルの強さにまで達している．チームとしての強さに焦点を当てた研究としては，Sander,B. らの研究 [3] で，QuakeIII のゲーム AI を動的に敵チームに適応させることで強いチームを生成する手法が提案されている．これら，強いゲーム AI の作成を目的とした研究は，人間プレイヤの「手強い敵」として人間を楽しませるための研究といえる．

しかし，人間プレイヤの「仲間」として強さ以外の面で人間プレイヤを楽しませるゲーム AI の研究は少ない．市販のコンピュータゲーム特に RPG と呼ばれるジャンルでは，ゲーム AI が操作するキャラクタとチームを組んで遊べるものも多いが，しばしば仲間 AI プレイヤは期待に反する行動を取り，プレイヤの不満に繋がる．これはこの種のゲームに“勝つ”以外の副目的が複数あり，AI プレイヤは人間プレイヤの“どの目的をどの程度重視しているか”といった価値観を理解せずに行動していることが原因の一つである．本研究では，人間プレイヤが選択した行動から人間プレイヤの重視する目的を推定し，それを AI プレイヤの行動選択に活用することでその人間プレイヤにとって満足度が高い AI プレイヤを生成することを目指す．

第2章　関連研究

ゲーム AI の研究には，単に強くする以外にも，挙動を自然にしたり人間と遊んで楽しめる AI の開発を目的とした研究がある．挙動の自然さについては例えば藤井らによる研究 [7] があり，挙動を学習する AI に生物の身体的制約を模した制限を課すことで人間らしい振る舞いの学習を目指す手法を提案し，アクションゲームの “Infinite Mario Bros.” において実装と評価を行った．Matteo ら [1] はゲーム AI の行動群に感じられる自然さとして Believability という指標を与えて，そのためには例えばある目的に沿う一貫性のある行動をとらせる必要があるとした．

チームプレイ AI の研究としては Sander らの研究がある．彼らは既存の AI が多く持つ特徴である，相手の戦略に関わりなく一貫して自分の戦略を用いる点を問題とした．QuakeIII のゲーム AI に進化的アルゴリズムに基づいた手法を適用して，相手の戦略に動的に対応しながら自らの戦略を相手に対して有利なものに変える挙動を AI にとらせて強さの向上を図った [3]．

我々が着目するのはチームプレイが必要なゲームの中でも，現在日本での人気が高いコマンド形式 RPG である．こうしたゲームでは戦闘に勝利するということが主目的であるものの，さらに “ どのような勝利を目指すのか ” といったことも重要になる．より圧倒的な勝利がプレイヤにとって望ましいことは将棋や前述の QuakeIII のようなゲームでも同じかもしれないが，コマンド形式 RPG では多くの場合，プレイヤ側が有利な状況にある非対称性を持つかわりに複数の戦闘が連続して発生し，1 回の戦闘の終了状態が次の戦闘の開始状態に一部引き継がれる点が大きく異なる．そのため，例えば「キャラクタをできるだけ傷つけない」「キャラクタの魔法力をなるべく温存する」「できるだけ早く勝利する」といった副次的な目的（以下，副目的）が重要になってくる．

Sander らが提案した手法は敵の行動へ適応し高い勝率の達成を目的としているが，我々は味方プレイヤの行動から，そのプレイヤの勝利以外の副目的を読み取りそれを満足させるような AI の作成を目指す．そのためには人間プレイヤの価値観をなんらかの形でモデリングする必要があると考える．

一般的なプレイヤの行動のモデリングという観点では様々な機械学習法が用いられ，それが α β 探索 [5] やモンテカルロ木探索 [2] を効率化することが広く知られている．また個別のプレイヤをモデリングすることで相手の弱点につけいることもしばしば行われる [6][8]．

あるいはいわゆる状態（盤面）評価関数の学習はプレイヤの価値観のモデリングとも言え，これも一般的なプレイヤまたは特定のプレイヤに対するものが広く用いられている

[9][11]．我々の研究では想定ゲームの「副目的が複数存在する」「状態遷移が確率的」といった特性を踏まえ，これらをうまく扱える効用の概念を用いて価値観のモデリングを試みる．

効用とは個々人が持つ，物事への価値観のことである [4]．例としてある個人が冷蔵庫を買う場面を考える．その人の冷蔵庫に対する効用の大きさが関数として $(w_1 * \sqrt{(\text{貯蔵容量})} + \frac{w_2}{(\text{値段})})$ で定義されるとすれば，数ある冷蔵庫の中から彼はこの値が最大となるような冷蔵庫を選んで購入する事になる．極端に言えば，この w_1 が 0 である人間がいればその人は値段が一番安い冷蔵庫を選んで買い，逆に w_2 のみが 0 となる人がいればどれほど高くても容量の大きな冷蔵庫を買う．またこの効用関数は非線形である．第一項はルートのオーダであり，傾きが次第に緩やかになる．これは一定以上の貯蔵容量があれば，それ以上の容量の増大にその人があまり魅力を感じないという事を示している．このように効用を関数として定めればその人間の選択する行動と価値観の間にある程度の説明ができる．

このような効用モデルを用いて吉谷らは，コマンド形式 RPG において人間プレイヤの意図や価値観を学習させる仲間ゲーム AI の構成方法を提案した [12]．しかしこの論文の設定と手法では事前に人間プレイヤが 100 回もの戦闘を同じ敵に対して行う必要があり，これは現実での利用を考えると好ましくない．そこで本研究では問題の単純化と手法の改良により早い段階での学習を目指した．

第3章　着眼点とアプローチ

本研究の目的は「人間が一緒にプレイして満足度が高い（ストレスや不満が少ない）仲間」の AI プレイヤを構成することである．本研究で扱うゲームでは仲間 AI プレイヤと人間プレイヤがチームを組んで遊ぶことができ，チームのメンバーは適切に協力しながらゲームを進めることが重要となる．

しかしこのようなゲームで仲間 AI プレイヤはしばしば人間プレイヤの価値観に沿わない，不愉快な行動をとる．その原因を吉谷らは

(a) 人間プレイヤと仲間 AI プレイヤの目的や価値観が異なる結果，それぞれにとっての最適な行動も異なってしまうため

(b) 目的は同じであるが，仲間 AI プレイヤの探索アルゴリズムなどが不十分で最適化行動がとれないため

(c) 効用関数は同じであるが，人間プレイヤ側の力量によって最適な行動を把握できず，“間違った”不満を抱いているため

(d) 本質的に混合戦略が必要なゲームであり，一定確率で期待と違ってもしょうがないため

の 4 つに分類している [12]．本研究でも (a) の原因に注目し，仲間 AI プレイヤに人間プレイヤの効用を学習させるというアプローチをとることで，行動選択の差異による人間プレイヤへのストレスを軽減させる．

アプローチの全体像を図 3.1 に示し，アプローチの流れを以下に示す．

1. ゲーム設定：本研究では，複数キャラクタによる味方チーム及び敵チームが存在するゲームを想定する．適用対象をコマンド形式 RPG には限定しないが，離散的なタイムステップと行動空間を持つゲームを想定する．1 回の戦闘毎に勝利を目指すが，勝利した場合でもその最終状態が好ましい場合と好ましくない場合があり，しかもそれが点数などで明確でないことを想定する．この前提はコマンド形式 RPG では珍しくない．

2. プレイヤ行動の記録：人間プレイヤが行動を選択し，それを仲間ゲーム AI が観測し記録する．このとき，どういう状態でどの行動を選択したかという対を，価値観推定のための参考データとして保存する．

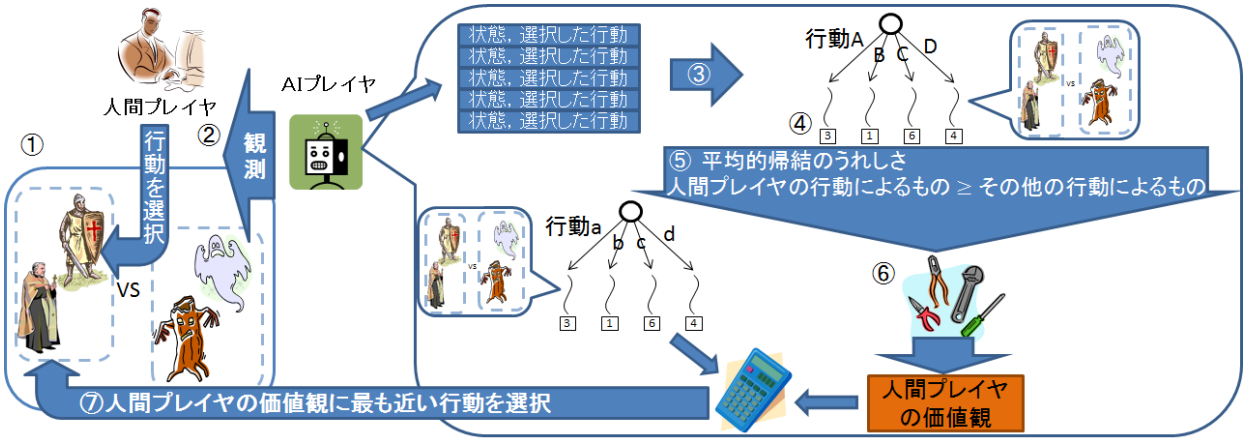

図 3.1: アプローチの全体像

3. 価値観の推定開始：人間プレイヤの行動が溜まってくると，それをもとに価値観の推定を開始する．

4. 平均的帰結のシミュレーション：人間プレイヤの選択した行動が何を目指したものなのかを推定するため，人間プレイヤが取りえた各行動を選択していた場合，対戦結果がどのようなものになりそうなのか,「この行動をとったらこうなりそう」という**平均的帰結**を求める．そのために，各行動後のゲームの推移を複数回シミュレーションし，様々な対戦結果を平均するアプローチをとる．

5. 帰結の解釈：人間プレイヤは副目的を達成するため，ある帰結へ向かうのに最適な行動を選択するはずである．それはつまり,「人間プレイヤのとった行動の帰結は，その他の行動の帰結と比べて人間プレイヤにとって嬉しい帰結である」と解釈できる．

6. 価値関数の推定：各プレイヤは「どんな対戦結果をより好ましいと感じるか」を決める独自の効用関数を暗黙に持っていると仮定できる．そこでその効用関数を近似するため，ゲームの結果を引数とし効用値を戻り値とする何らかの関数モデルを作成し，可変パラメータを持たせる．そのうえで，(5) で示した条件ができるだけ満たされるように効用関数のパラメータを最適化する．

7. 人間プレイヤが満足する行動を決定：推定した人間プレイヤの価値観で最善となるような行動を選択することで，人間プレイヤに迎合する．

第4章　提案手法

本章では，前章で述べたアプローチを具体的にどのように実装したか，詳細を図3.1の数字順に説明する．本論文で用いる記号は登場するごとに説明するが，それらをまとめたものが表4.1である．

4.1　プレイヤ行動の記録

本論文で扱う対象は，離散の状態集合 S，離散の行動集合 A を持つマルコフ決定過程とする．ある人間プレイヤの価値観を推定したい場合，その人間プレイヤの行動履歴を蓄積し，推定に用いる．人間プレイヤの j 回目の行動選択時の状態を $s^j \in S$，その時の合法手を $A_{s^j} \subset A$，その時選択した行動を $a^j \in A_{s^j}$ と書くと，行動履歴は状態と行動の対集合，$\{(s^j, a^j)\}^j$ と書ける．

表 4.1: 本論文で登場する記号

$s \in S$	現在の状態
$A_s \subset A$	状態 s での全合法手
$a \in A_s$	合法手
$a^* \in A_s$	人間プレイヤが選んだ手
$\pi : S \to \mathbb{R}^{\lvert A \rvert} \in \Pi$	戦略
$s_i(s, a, \pi)$	状態 s から初手 a，以降戦略 π でシミュレーションした i 回目の結果の状態
$\vec{x}_i(s, a, \pi) \in \mathbb{R}^n$	状態 $s_i(s, a, \pi)$ の特徴量ベクトル
$\vec{x}(s, a, \pi) \in \mathbb{R}^n$	$\{\vec{x}_i(s, a, \pi)\}_i$ の平均値
$\vec{w}$	効用関数の重み
$u(\vec{x}, \vec{w}) \in \mathbb{R}$	効用関数の重み $\vec{w}$，特徴量ベクトル $\vec{x}$ の時の効用値

4.2 平均的帰結のシミュレーション

状態 s で行動 $a^* \in A_s$ が選ばれた 1 つの履歴について考える．この a^* または別の行動 $a \in A_s$ を選択した場合に，どのような帰結を迎えるかをモンテカルロシミュレーションによって推定したい．シミュレーションは，状態から行動の確率分布への写像である戦略 $\pi : S \to \mathbb{R}^{|A|}$ に基づいて行う．状態 s，行動 a，戦略 π による i 回目のシミュレーションの結果を $s_i(s, a, \pi)$ と書く．

状態 $s_i(s, a, \pi)$ そのままでは非常に多くの情報を含むため，そこから特徴量抽出関数 $S \to \mathbb{R}^n$ を用いて実数値ベクトル $\vec{x}_i(s, a, \pi)$ を得ることにする．さらに，m 回のシミュレーションの結果を線形に平均することで，平均的帰結を表すベクトル

$$\vec{x}(s, a, \pi) = \frac{1}{m} \sum_{i=1}^{m} \vec{x}_i(s, a, \pi) \tag{4.1}$$

を求める．

以上の平均的帰結ベクトル $\vec{x}$ を求めるシステムを本論文では効用要素推定器と呼ぶ．図 4.1 にその概念図を示す．これは深さ 1 モンテカルロ法であるが，本論文の実験では，シミュレーションに複数の戦略 π を用いて，戦略ごとに異なる帰結が導かれることを想定している点が特徴的である．

完全にランダムなシミュレーションよりも“ありそうな行動”を高い確率で選択するように偏らせたシミュレーションの方がよいことが多いことは知られているが [2]，副目的のどれを重視するかによって“ありそうな行動”が全く変わってくるゲームにおいては適切な固定の戦略を定めることが困難である．そこで我々は複数の戦略を用意し，それぞれの行動それぞれの戦略ごとに平均的帰結 $\vec{x}(s, a, \pi)$ を求める．

プレイヤが行動 a_2 を取らずに a_1 を取ったとすれば，それは「ある戦略 π^* による平均的帰結 $\vec{x}(s, a_1, \pi^*)$ は全ての $\vec{x}(s, a_2, \pi)$ よりもそのプレイヤにとって優れている」と解釈できる．

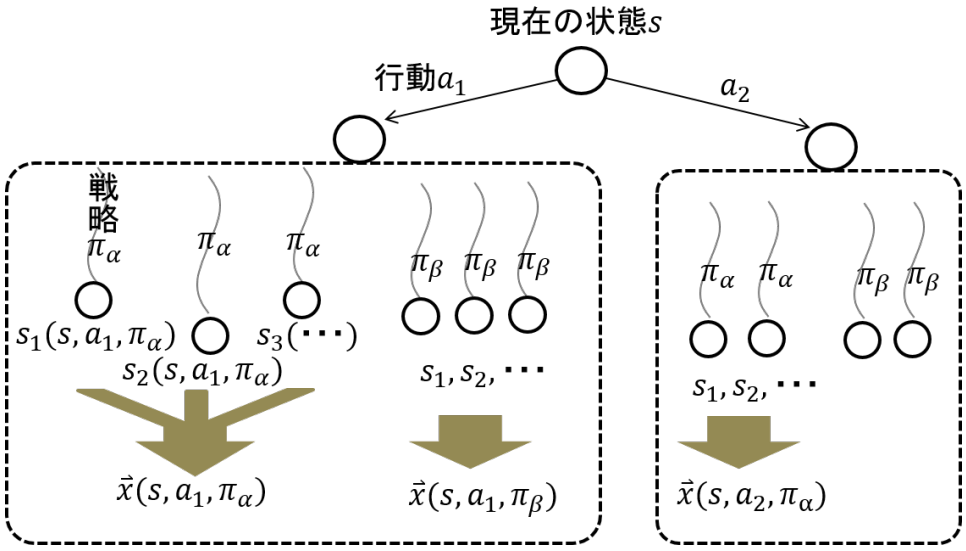

図 4.1: 効用要素推定器の全体像

4.3 価値関数の推定

効用要素推定器により得られた情報を基に人間プレイヤの価値関数を推定する．人間プレイヤがある効用関数に基づいて行動しているとしてもその正確な関数モデルは不明である．仮の関数モデルとして単純なものから複雑なものまでさまざまなものが利用できるが，本論文では単純な線型和モデルを用いることにする．我々が用いる効用関数 $u : \vec{x} \to \mathbb{R}$ を式 (4.2) に示す．

$$u(\vec{x}(s, a, \pi), \vec{w}) = \vec{x}(s, a, \pi) \cdot \vec{w} \tag{4.2}$$

ここで $\vec{x}(s, a, \pi)$ は状態 s と行動 a と戦略 π により平均的に到達する戦闘終了状態の特徴量ベクトルであり $\vec{w}$ は重みベクトルである．

人間プレイヤが行動 a^* を他の行動より優先して選んだことは，それによって得られる効用値が他の行動によって得られるものより大きいからと考えることができ，以下の不等式 (4.3) が成り立つことが期待される．

$$\max_{\pi \in \Pi} u(\vec{x}(s, a^*, \pi), \vec{w}) \geq \max_{\pi \in \Pi, \overline{a} \in A_s} u(\vec{x}(s, \overline{a}, \pi), \vec{w}) \tag{4.3}$$

ただし Π は可能な戦略 π の集合である．この式が満たされない場合は，その効用関数は不適切，つまり重みベクトル $\vec{w}$ は不適切である可能性が高いと考える．

そこで我々は，人間プレイヤのありうる重みベクトル空間 $W \ni \vec{w}$ を有限離散集合として定義し，各ターンにおけるプレイヤの選択行動 a*が観測されるたびに不等式 (4.3) を満たさないベクトル $\vec{w} \in W$ にペナルティを与える．そしてその人間プレイヤの持つ効用関数の重みベクトルは，候補となる全重みベクトル $\vec{w} \in W$ のうちペナルティの最も低いものであるとして推定を行う．具体的な処理の流れをアルゴリズム 1 に示す．

4.4　人間プレイヤが満足する行動を決定

前節までに，各行動aと戦略πから平均的帰結$\vec{x}(s,a,\pi)$を導く方法と，効用関数$u(\vec{x},\vec{w})$を推定する方法を述べた．これらを用いて人間プレイヤに迎合しようとする場合，最も期待効用値が高くなるように$\arg\max_{a\in A_s,\pi\in\Pi} u(\vec{x}(s,a,\pi)\cdot\vec{w})$を行動として選択すればよい．

Algorithm 1 プレイヤの重みベクトル推定アルゴリズム

for each $\vec{w}\in W$ **do**
  $p_{\vec{w}}=0$
end for
for each $(s,a^*)\in\{(s^j,a^j)\}^j$ **do**
  for each $\vec{w}\in W$ **do**
    $u*=\max_{\pi\in\Pi} u(\vec{x}(s,a^*,\pi),\vec{w})$
    for each $a\in A_s\setminus a^*$ **do**
      if $u*<\max_{\pi\in\Pi} u(\vec{x}(s,a,\pi),\vec{w})$ **then**
        $p_{\vec{w}}+=1$
      end if
    end for
  end for
end for
return $\arg\min_{\vec{w}\in W} p_{\vec{w}}$

第5章　本研究で扱うゲームの概要

本研究では独自にコマンド形式のターン制ゲームを設計し，提案手法の実装や実験を行う．このゲームはマルコフ決定過程で記述できる多人数順次着手確定ゲームである．ゲームの各種設定は以降に説明する．

5.1　キャラクタの持つパラメータ

ゲームに参加するキャラクタには以下の5つのパラメータを設定する．キャラクタを操作するプレイヤは，自身が操作しているキャラクタのパラメータだけでなく，敵・味方全てのパラメータを知ることができる．市販ゲームではパラメータの一部が不完全情報になることが多いが，プレイを進めるにつれて同種の敵のパラメータはプレイヤが把握できるようになる．本論文ではこの把握の過程を省き課題を明確にするため，完全情報として扱うことにした．

- 体力（HP）: 体力が0になると，そのキャラクタは行動不能になる．チーム全員の体力が0になった場合，そのチームは戦闘に敗北する．可変値．
- 精神力（MP）: 一部の術技（行動の種類）を使用するために必要で，対象となる術技を使用すると値が減少する．可変値．
- 攻撃力：攻撃をした際，相手に与えるダメージの計算に使用される．固定値．
- 守備力：攻撃を受けた際，自分が受けるダメージの計算に使用される．固定値．

5.2　行動とその効果

各キャラクタの行動は，術技と対象キャラクタの組み合わせで表現される．各術技の効果は以下の通りである．

- 単体攻撃：対象キャラクタに（自身の攻撃力 - 対象キャラクタの守備力）の値をダメージとして与える．
- 回復：精神力を一定量消費して，対象キャラクタの体力を一定量回復する．なお，精神力を消費する技を使うキャラクタに消費される分の精神力が残っていない場合は選択できない．
  - 小回復：精神力を 4 消費して対象キャラクタ 1 体の体力を 42 回復させる．
  - 中回復：精神力を 8 消費して対象キャラクタ 1 体の体力を 88 回復させる．
  - 全体回復：精神力を 18 消費して味方キャラクタの体力を全回復させる．
- グループ攻撃：同じ種類の敵数体はグループを組むが，精神力を 8 消費して，その 1 グループ全体に一定のダメージを与える．
- 防御：次に自分の番がくるまで受けるダメージを半減する．

以上のように行動は最大で 6 種類，10 通りほどになる．

5.3　状態遷移

このゲームでは複数のキャラクタが敵チーム・味方チームに分かれ，チーム同士で戦闘を行う．ゲームの流れを図 5.1 に示す．ゲーム進行の単位としてターンが存在し 1 ターンの中で全てのキャラクタが行動するが，行動する順番はランダムである．行動の結果は直ちに反映される．戦闘不能になったキャラクタは行動の順番は回ってこず，味方か敵が全滅するまで上記を繰り返す．市販の RPG ゲームでは，ターンの開始時に全ての味方の行動を決定しなければいけない場合も多く，その場合にはナッシュ均衡や混合戦略などより複雑な処理が必要になる．吉谷らの研究ではこのようなゲームを扱ったことも学習に多くのデータを必要とした原因であり，我々はまず単純な状態遷移ルールのゲームを扱う．

図 5.1: ゲームの流れ

第6章　設定と戦略

実験で使用する設定とモンテカルロシミュレーションで使用する戦略について説明する．

6.1　戦闘参加キャラクタの設定

このようなゲームでは敵味方のキャラクタの能力は非対称であり，それによってとるべき行動や効用推定の難しさも異なるため，我々は評価のために5つの多様な設定を準備した．戦闘参加キャラクタの設定を表6.1，A.1〜A.4に示す．ここでは，設定2について説明する．設定2は，小回復ができる敵を含む敵チームを倒すという設定である．速度重視であれば攻撃力の高いグループ攻撃が有効な攻撃手段となるが，これはMPを消費するためMP重視の効用の場合は控えなければならない．さらに敵の攻撃力が高いためHP重視の場合は回復にも気を配る必要がある．

表 6.1: キャラクタのパラメータ　設定2

キャラクタ	HP	MP	攻撃力	守備力	使用可能術技
味方1	134	30	60	28	単体攻撃・小回復・防御
味方2	108	60	44	34	単体攻撃・グループ攻撃 ・小回復・中回復 ・全体回復・防御
敵1	52	0	40	26	単体攻撃
敵2	82	32	38	32	単体攻撃・小回復
敵3	70	0	50	30	単体攻撃

6.2　取りうる戦略の設定

モンテカルロ法におけるシミュレーション戦略を複数用意した点は本論文の特色の一つである．本論文では以下に示す 7 つの戦略を実験に用いた．

1. ランダム：全ての合法手を等確率で選ぶ．
2. 適時回復：HP 回復を行う手は味方の残り HP が高いほど選択確率を低くする．
3. 攻撃重視：単体攻撃・グループ攻撃の選択確率を 5 倍にする．
4. 単体攻撃重視：単体攻撃の選択確率を 5 倍にする．MP を節約したい場合に適する．
5. グループ攻撃重視：グループ攻撃の選択確率を 5 倍にする．
6. 単体＋適時回復：(2) と (4) を混合したもの．
7. グループ＋適時回復：(2) と (5) を混合したもの．

6.3　予備実験 1 ：戦略の価値

本節では，効用関数が試合結果に与える影響，戦略を複数与えることの価値を調べるための予備実験を紹介する．実験はボス戦を想定したやや厳しい設定 5 で 1000 戦した．味方キャラクタ 1 はルールベースで動作させ，味方キャラクタ 2 は「HP 重視」と「MP・Turn 重視」の効用関数を持つモンテカルロ AI とした．その結果を表 6.2 に示す．ここで完全勝率とは全戦闘における味方キャラクタが全員生き残った戦闘の割合のことである．

この結果から，キャラクタに与えた効用関数によって試合結果が有意に変わりうること，また戦略がない場合はある場合に比べ有意に試合結果が悪いことが分かる．

表 6.2: 戦略がない場合の勝率の変化

	戦略がある場合		戦略がない場合	
	勝率	完全勝率	勝率	完全勝率
HP 重視	99%	99%	96%	96%
MP・Turn 重視	98%	83%	93%	69%

このように戦略を複数用意するとモンテカルロ AI は強くなる．ある程度強いシミュレーションでなければ効用要素の正しい推定（平均的帰結の測定）もできず，推定した効用関数を用いてプレイヤにうまく迎合できないと考える．実際，8 章で説明する実験を戦略なしで行うと，迎合に成功する率が 10% 以上悪くなることが分かっている．

6.4　典型的な行動の分岐点における評価値の違い

本節では，典型的な行動の分岐点における評価値の違いを 3 例紹介する．これらは一見副目的と実行した行動が矛盾しているように見えるが，実は理にかなった行動を行っている例であり，副目的に沿った行動を行うという問題は複雑な問題であるということを説明する．

評価値が一見合理的に見えるが実は合理的ではない行動には低い値，一見合理的ではないように見えるが実は合理的な行動には高い値が与えられていることに注目してほしい．さらに同じ行動でも戦略が異なるだけで評価値に大きな違いが生まれていること，副目的により評価値の高い戦略は異なることにも注目してほしい．

なお，全ての例で行動を起こそうとしているのは味方 2 である．

6.4.1 HP が少ない時 HP 優先であるのに単体攻撃を行う例

1 例目は，HP が少ない時 HP 優先であるのに単体攻撃を行う例である．その時の状態を表 6.3 に示し，代表的な行動の評価値を表 6.4〜6.6 に示す．太字となっている値が最も重要な値であり，最大値は斜体である．味方全員の HP を全回復できる全体回復を使うよりも敵 2 に単体攻撃をするときの評価値のほうが高い．これは今 HP を回復するよりも高い攻撃力を持つ敵 2 を倒したほうが HP を高く保てるからである．この状態から敵 2 を倒さずに HP の回復を優先した場合，敵 2 から最低でも 46 のダメージを受ける．敵 2 を倒さなければこのダメージを何回も受けなければならない．一方，HP の回復をせず敵 2 を倒した場合，敵から受けるダメージは最高でも 12 となる．つまり，敵 2 を倒したほうが HP の減りが少なるため敵 2 に単体攻撃を行うという結論となる．

表 6.3: HP 優先であるのに単体攻撃を行う時の状態

	HP/Max	MP/Max	攻撃力	守備力
味方 1	118/142	28/32	66	36
味方 2	36/112	64/64	48	38
敵 1	60/60	0/0	48	26
敵 2	28/84	0/0	84	20
敵 3	84/84	0/0	44	60

表 6.4: HP 優先：敵 2 に単体攻撃時の評価値

戦略	重み		
	HP	MP	Turn
1	72.7	0.00	4.85
2	77.1	24.2	39.0
3	61.6	33.6	81.3
4	53.5	23.5	40.8
5	61.9	32.5	102
6	80.1	44.2	28.0
7	***83.5***	27.0	90.7

表 6.5: HP 優先：敵 1 にグループ攻撃時の評価値

戦略	重み		
	HP	MP	Turn
1	38.8	0.21	3.98
2	44.8	5.60	12.0
3	32.3	13.0	51.8
4	39.1	14.6	23.7
5	37.4	11.0	61.4
6	55.9	19.4	23.6
7	**62.2**	5.18	65.6

表 6.6: HP 優先:味方に全体回復時の評価値

戦略	重み		
	HP	MP	Turn
1	59.2	0	3.63
2	69.0	13.8	32.2
3	69.9	18.2	70.8
4	**79.0**	26.6	17.5
5	66.0	11.7	83.6
6	77.9	33.4	25.0
7	73.9	16.3	82.4

6.4.2 Turn 優先であるのに中回復を行う例

2 例目は，Turn 優先であるのに中回復を行う例である．その時の状態を表 6.7 に示し，代表的な行動の評価値を表 6.8〜6.10 に示す．この状態から HP の回復をせず敵 2 にグループ攻撃をした場合，敵 2 の次の行動で味方 2 が倒されることが予想される．こうなると味方 2 が敵に与えられたはずのダメージを味方 1 のみで与えなくてはならない．味方 2 のグループ攻撃は敵 3 に対して大きいダメージを与えられるため，味方 2 が倒れてしまった場合確実に戦闘が長引く．一方，敵 2 に攻撃せず HP を回復した場合，攻撃行動が 1 回休みになるが一撃で倒されることはないため，HP 回復後に再度攻撃に加わることができる．HP の回復もあまり HP の減っていない味方 1 にも回復を行う全体回復を行うより味方 2 だけ回復し，全体回復と中回復の差分をグループ攻撃に使うほうが有効である．これらより，攻撃を多く行うために中回復を行うという結論となる．

表 6.7: Turn 優先であるのに中回復を行う時の状態

	HP/Max	MP/Max	攻撃力	守備力
味方 1	118/142	24/32	66	36
味方 2	40/112	64/64	48	38
敵 1	0/60	0/0	48	26
敵 2	56/84	0/0	84	20
敵 3	84/84	0/0	44	60

表 6.8: Turn 優先 : 敵 2 にグループ攻撃時の評価値

戦略	重み		
	HP	MP	Turn
1	35.3	0.175	2.93
2	45.3	8.14	20.2
3	30.6	17.1	54.6
4	38.0	18.1	25.6
5	36.2	16.6	**67.4**
6	57.6	21.8	32.2
7	54.0	9.62	61.0

表 6.9: Turn 優先 : 味方 2 に中回復時の評価値

戦略	重み		
	HP	MP	Turn
1	56.4	0.204	3.97
2	60.6	15.0	23.0
3	54.7	29.6	71.6
4	60.1	30.3	20.7
5	57.8	27.3	***89.2***
6	73.3	36.5	19.6
7	74.0	21.9	77.3

表 6.10: Turn 優先 : 味方に全体回復時の評価値

戦略	重み		
	HP	MP	Turn
1	59.6	0.00	1.77
2	61.8	10.8	20.7
3	63.3	25.3	73.5
4	70.0	28.7	26.3
5	58.9	19.6	**83.0**
6	74.3	29.4	21.5
7	68.3	17.1	67.7

6.4.3 MP 優先であるのにグループ攻撃を行う例

3 例目は，MP 優先であるのにグループ攻撃を行う例である．その時の状態を表 6.11 に示し，代表的な行動の評価値を表 6.12〜6.14 に示す．この状態から敵 1〜3 にグループ攻撃をせず敵 2 に単体攻撃をした場合，敵を 1 体減らすことができるが，グループ攻撃を行った場合と違い次に行動する味方はどのような攻撃をしても敵をさらに 1 体減らすことはできず，敵から受ける総ダメージをグループ攻撃よりも増やす結果となり，余分に回復をしなければならないため使用する MP が増えてしまう．つまり，回復による MP の消費を抑えるためにグループ攻撃をするという結論となる．

表 6.11: MP 優先であるのにグループ攻撃を行う時の状態

	HP/Max	MP/Max	攻撃力	守備力
味方 1	114/134	30/30	60	28
味方 2	60/102	48/80	44	32
敵 1	52/52	0/0	38	26
敵 2	18/52	0/0	38	26
敵 3	52/52	0/0	38	26
敵 4〜6	0/52	0/0	38	26
敵 7	52/52	0/0	38	26
敵 8	0 /52	0/0	38	26
敵 9	52/52	0/0	38	26
敵 10	52/52	0/0	38	26

表 6.12: MP 優先：敵 2 に単体攻撃時の評価値

戦略	重み		
	HP	MP	Turn
1	65.2	5.72	55.2
2	50.5	14.8	76.7
3	18.5	24.2	93.7
4	23.2	27.6	85.6
5	31.4	**31.3**	148
6	61.8	26.7	90.9
7	69	18.7	117

表 6.13: MP 優先：敵 1〜3 にグループ攻撃時の評価値

戦略	重み		
	HP	MP	Turn
1	73.9	11.3	86.5
2	66	22.8	105
3	33.5	36.2	145
4	33.7	***41.7***	138
5	47.1	33.2	163
6	70.3	32.9	119
7	74.3	21.1	142

表 6.14: MP 優先：敵 9〜10 にグループ攻撃時の評価値

戦略	重み		
	HP	MP	Turn
1	51.4	2.53	56.9
2	39.1	10.3	70.2
3	17.2	22.3	99
4	11.5	13.5	53.1
5	31.8	**29.3**	153
6	52.8	21.1	86.2
7	64.1	15	127

このように「MP を節約したいなら MP を消費しない行動をとりつづければいい」という単純な問題ではなく,「MP を節約したいならここで MP を消費するこの行動を行わなければならない」というように，副目的と一見矛盾した行動をとることが起こりうる複雑な問題である．

第7章　シミュレーション結果の特徴量化

本章では，シミュレーションの結果をどのように特徴量化するのか，また効用関数の重みベクトル $\vec{w}$ が変化するとどの程度選択される行動が変わってくるのかを説明する．

7.1　状態評価関数

シミュレーション結果には様々な情報が含まれているが，そこから効用要素となる情報を抜き出し 3 次元の特徴量ベクトルとした．この情報が多ければ多いほどより詳細に効用関数をモデリングできるが，その分必要になる計算量・メモリ・データ数も増える．

平均的帰結 $\vec{x}$ が持つ特徴量を式 7.1 に示す．

$$\vec{x} = \{x_{HP}, x_{MP}, x_{Turn}\} \tag{7.1}$$

式 7.1 に示された各特徴量の計算式を式 7.2〜7.4 に示す．ここで，a, b, は設定ごとに異なる定数である．

$$x_{HP} = \frac{\text{自チーム全体の残り HP}}{\text{自チーム全体の最大 HP}} \tag{7.2}$$

$$x_{MP} = \frac{\text{自チーム全体の残り MP}}{\text{自チーム全体の最大 MP}} \tag{7.3}$$

$$x_{Turn} = b - \text{経過ターン数} \times a \tag{7.4}$$

7.2　重みベクトル空間

効用関数の重みベクトル $\vec{w}$ は $\vec{x}$ と同様3次元ベクトルであり，それぞれ x_{HP}, x_{MP}, x_{Turn} に対応する．本研究で用いるモデルは線形重み和であるため1次元目は1に固定できる．例えば [1,10,0.1] は MP 重視，[1,0.1,0.1] は HP 重視などと考えることができる．

本研究では人間プレイヤのありうる重みベクトル空間 W を x_{MP}, x_{Turn} に対応する2次元のマトリクスとし，大きさを 31×31 としたが，重みはログスケールであり，各重みの最大値は32，最小値は $\frac{1}{32}$ である．

7.3　予備実験２：行動一致率の分布

効用関数の重みベクトル $\vec{w}$ の変化により選択される行動がどの程度変わるのかを実験により確認した．2 つの重みでモンテカルロ AI により選択される行動が一致する割合を“行動一致率”と定義する．ここで，全く同じ重み同士であっても，モンテカルロ法の乱数性により行動が一致するとは限らないことに注意されたい．

$\vec{w} = [1, 4, 8]$（MP と Turn 重視）に対する行動一致率の分布を図 7.1，$\vec{w} = [1, \frac{1}{8}, \frac{1}{16}]$（HP 重視）に対する行動一致率の分布を図 7.2 に示す．

$[1, 4, 8]$ の場合は線状に行動一致率の高い分布があり，その線を少しずれるだけで行動一致率は 10% 以上低下する．逆に $[1, \frac{1}{8}, \frac{1}{16}]$ の場合は面状に行動一致率の高い分布があり，$[1, 4, 8]$ より重みのずれに寛容である．このように行動一致率の分布の形は重みによって大きく変化する．

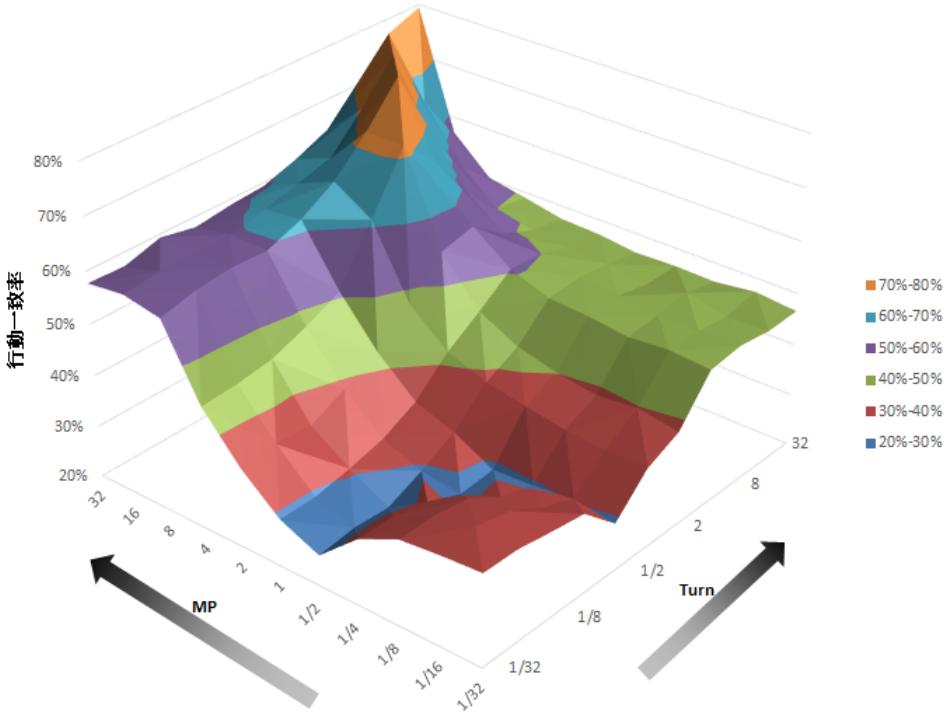

図 7.1: $[1, 4, 8]$ の場合の行動一致率の分布

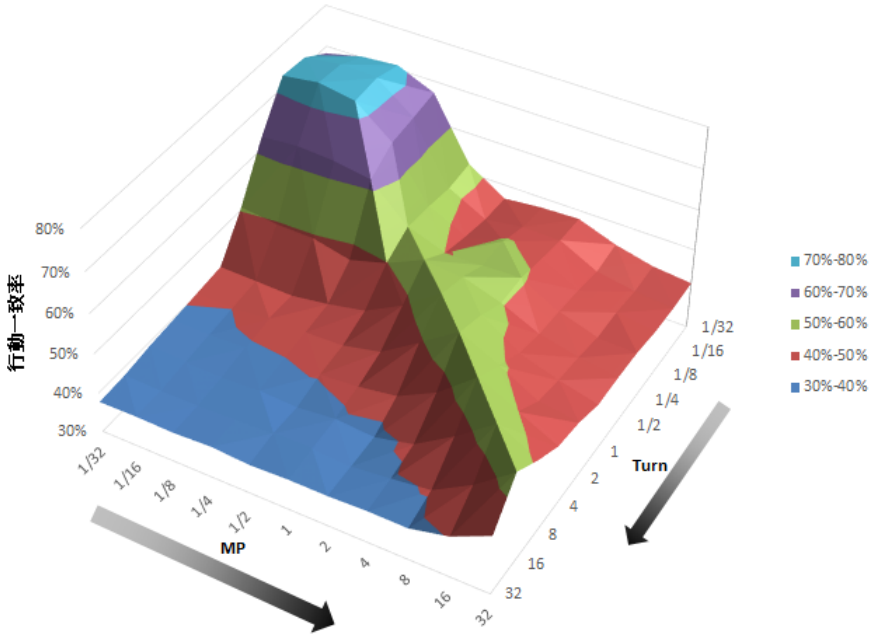

図 7.2: $[1, \frac{1}{8}, \frac{1}{16}]$ の場合の行動一致率の分布．前図とは各軸の向きが逆．

第8章　実験

本章では，特定の効用関数を持たせたAIの効用重みを提案手法が正しく推定できるか，および実際の人間に対して満足度を高めることができているか確認する．

8.1　人工プレイヤに対する学習実験

本節では，特定の効用関数を持たせた人工プレイヤを用意し，提案手法がそれを正しく学習できるか確認する．持たせた効用関数は表8.1に示す4通り，また敵味方の能力も5通り，全20パターンを調べた．RPGはプレイヤごとに価値観が異なり，想定される状況も多いため，このように複数のパターンを用いることにした．

味方1はルールベースの単純な動作をし，味方2は特定の効用関数を持たせた人工プレイヤが操作する．提案手法は味方2の挙動のみを記録し，その効用関数を推定することにする．戦闘は8回連続して行われ，1回目終了時点と8回目終了時点では推定の精度がどのように変わるかも調べる．戦闘そのもの・またモンテカルロシミュレーションには乱数が大きく影響を与えるため，この8回の戦闘を1セットとし，シード値を変えた20セットの学習を行う．

評価には，推定した効用関数の重みそのものではなく，それによって取る行動が元の効用関数によるものとどれだけ一致するかを用いる．これは，例えば図7.2において元の重み [1, 1/8, 1/16] とは異なる [1, 1/16, 1/32] のような重みとなったとしても行動一致率はさほど悪化しないつまり満足度を下げない場合があるためである．

図8.1には，持たせた効用関数の重みごとの一致率の推移を表す．横軸は戦闘回数で，1戦目の半分を経過した時点，1戦後・3戦後・5戦後・8戦後での行動一致率を表す．右端の点は，全く同じ効用関数を持たせた場合であり，いわば学習の限界点を意味する．この図では敵味方の能力5通りごとの結果は全て平均されている．例えば黄色の線と点の場合，[1,12,0.167] というMP重視の効用関数は，どのような状況においてもほぼ正しく早期に推定できて，一致率も限界に近い性能を出していることが分かる．なお，4つの効用重みの間に一致率15%ほどの差があるのは，「MPを消費せず勝ちたいなら，適当な敵を単体攻撃するか防御する」「早く勝ちたいならグループ攻撃をするしかない」などのように好ましい行動の幅が変化するためである．

表 8.1: 使用した効用関数の重み

	HP	MP	Turn
HP 重視	1	0.071	0.071
Turn 重視	1	0.143	18
MP と Turn 重視	1	10	10
MP 重視	1	12	0.167

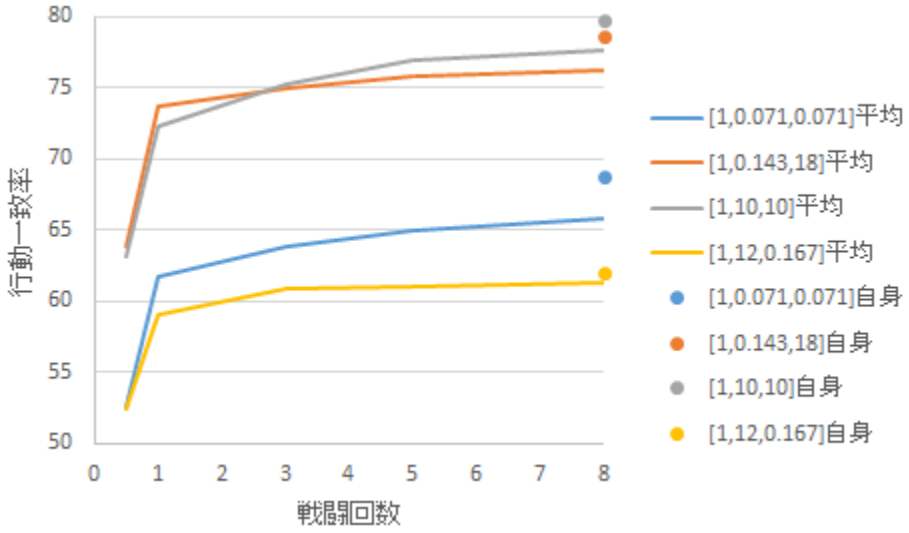

図 8.1: 行動一致率・効用関数の重み別

図 8.2 は逆に，5 つの設定ごとの一致率の推移を表す（与えた 4 つの効用重みに関しては平均化してある）．ここから，与えた状況ごとに多少得手不得手はあるものの，概ね早期に効用関数の推定ができていることが分かる．

図 8.3〜8.8 は推定した効用関数の重みの推移を示す．横軸が MP の重み，縦軸が Turn の重みである．図の順番に 1 戦目の半分を経過した時点，1 戦後・8 戦後での効用関数の重みの分布を示す．図 8.3〜8.5 は [1, 0.071, 0.071] という HP 重視の効用関数の重みの結果であり，5 つの設定をまとめて表示している．図 8.6〜8.8 は [1, 12, 0.167] という MP 重視の効用関数の重みの結果である．1 戦目の半分を経過した時点ではうまく推定できていないが，1 戦後には正しい方向へ推定が進んでおり，戦闘を重ねるにつれてそれぞれ [1, 0.071, 0.071]，[1, 12, 0.167] に近づいていることを示している．

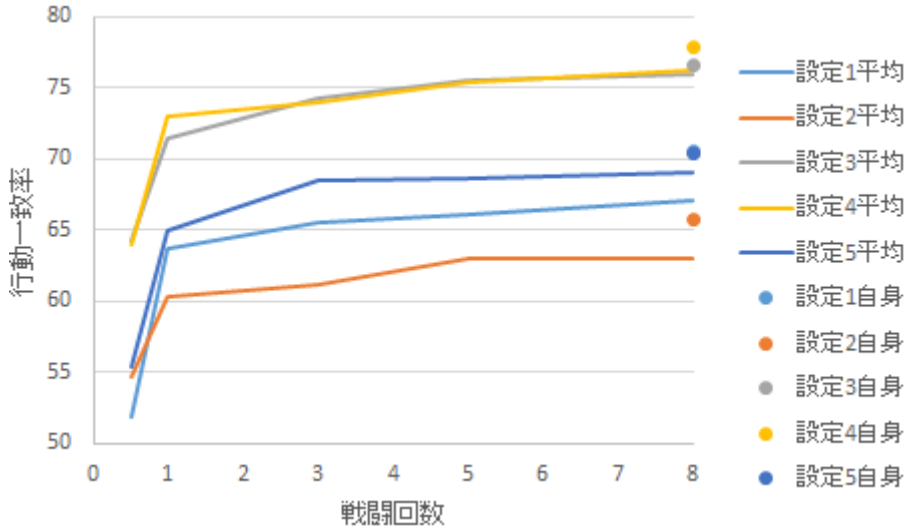

図 8.2: 行動一致率・設定別

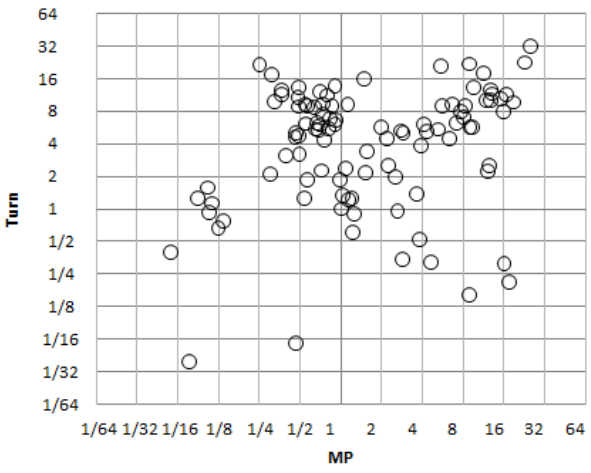

図 8.3: HP 重視 半戦後の効用重み分布

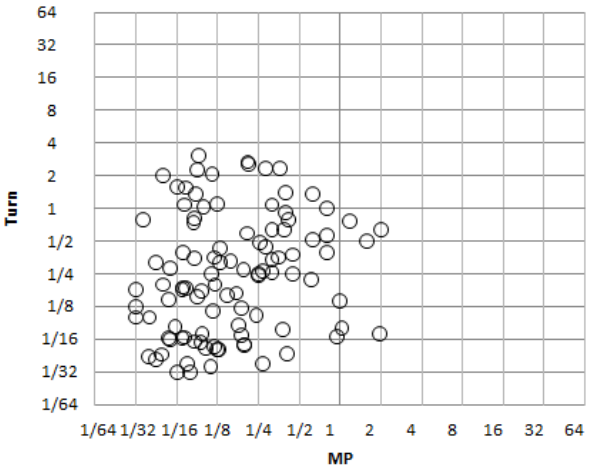

図 8.4: HP 重視 1 戦後の効用重み分布

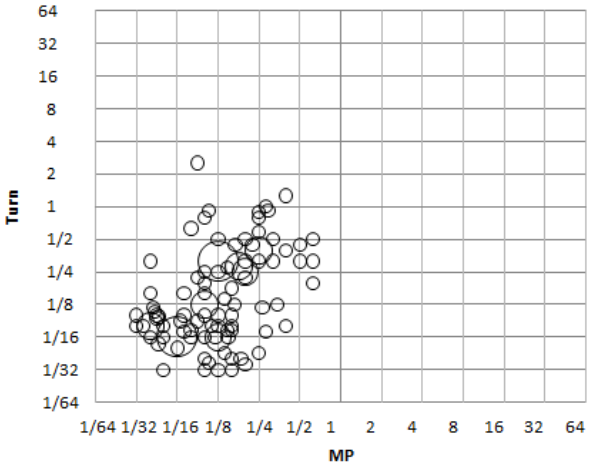

図 8.5: HP 重視 8 戦後の効用重み分布

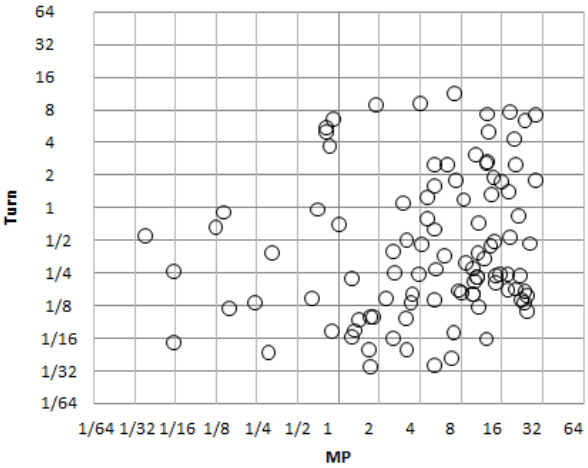

図 8.6: MP 重視 半戦後の効用重み分布

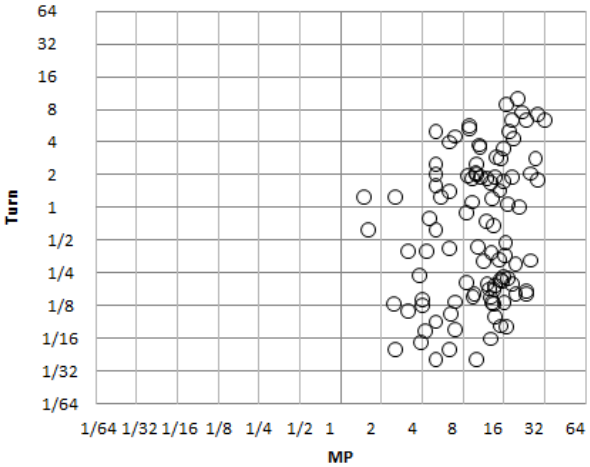

図 8.7: MP 重視 1 戦後の効用重み分布

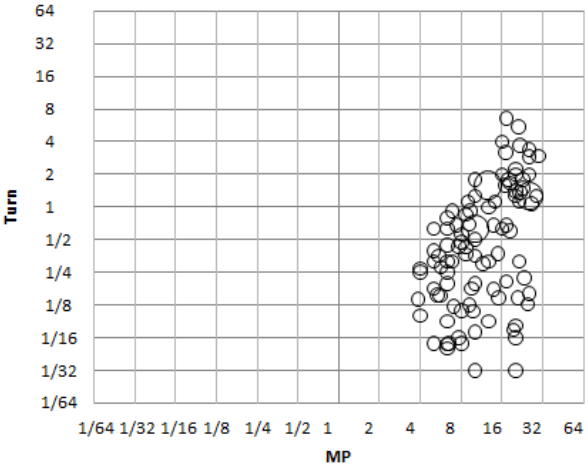

図 8.8: MP 重視 8 戦後の効用重み分布

8.2 被験者実験

前節の実験では，学習対象となる人工プレイヤは,「学習に用いる効用モデルと同じ効用モデル」と「学習に用いる行動決定アルゴリズム（戦略付きモンテカルロ法)」を持つものであり，いわば学習者にとって理想的な条件であった．

そこで，実際の人間プレイヤに対してどの程度学習ができるのかを確認するための被験者実験を行った．被験者にはまず戦闘を 2 回行ってもらい，対象ゲームに慣れてもらった．次に戦闘 8 回を 1 セットにして，3 セットの戦闘を行ってもらった．各セットでは被験者に「MP 重視」「Turn 重視」「MP・Turn どちらも重視」して戦闘を行うように指示した．

1 セットの中で前半の戦闘 4 回は学習フェイズで，被験者に味方キャラクタを全て操作してもらう．このとき味方キャラクタ 2 の操作を提案手法により学習する．1 セットの中で後半の戦闘 4 回は評価フェイズで，被験者に味方キャラクタ 1 のみ操作してもらい，AI が操作する味方キャラクタ 2 の挙動を 1 戦ごと，5 段階で評価してもらった．

評価してもらった AI は 3 通りで，一つは提案手法（効用関数を推定した AI）を 2 回，一つは固定の効用重み [1, 0.3, 3] を持つ Turn 重視の AI，一つは固定の効用重み [1, 4, 0.25] を持つ MP 重視の AI である．AI の登場順評価順はランダムである．

被験者 7 人による自然さの評価値の平均を表 8.2 に示す．例えば Turn 重視を指示した場合，Turn 重視 AI に対する評価 (4.0) は MP 重視 AI に対する評価 (2.6) よりも高いが，提案手法はそれ以上の評価 (4.3) となった．学習された重みは [1, 0.5, 16] などより極端なものであり，固定で与えた [1, 0.3, 3] では不十分だったことを示唆している．人手で効用関数を設計することは困難な場合が多く，提案手法のように行動から自動で推定することに価値があることが示せた．

表 8.2: 自然さの評価結果

指示したスタイル	使用 AI と重み	自然さの平均評価値
MP 重視	提案手法	4.1
	Turn 重視 AI	3.1
	MP 重視 AI	3.4
Turn 重視	提案手法	4.3
	Turn 重視 AI	4.0
	MP 重視 AI	2.6
MP と Turn 重視	提案手法	4.0
	Turn 重視 AI	3.0
	MP 重視 AI	2.7

第9章　発展的なペナルティ加算法の提案

前章までは章で説明した通り，不等式 (4.3) が不成立となる毎にペナルティを 1 加算していた．本章ではこの方法をさらに発展させた方法について述べる．

9.1　学習を高速化するペナルティの与え方

ペナルティは人間プレイヤの効用関数の重み（以降目標の重み）では不適切である可能性がある重みベクトル $\vec{w}$ を目標の重み候補から徐々に外していくものであった．この考えを推し進めれば,「これは目標の重みでない可能性が高い」という場合は加算するペナルティを大きくし,「これは目標の重みでない可能性が少しある」という場合は加算するペナルティを小さくしたほうが早く学習できるという結論に達する．

では，不等式 (4.3) が不成立になったときの右辺と左辺の差分を加算するペナルティに反映させればいいかというとそうではない．例えば不成立となった $\vec{w}_1$ と $\vec{w}_2$ の評価値を調べたところ次のようであったとする．

- $\vec{w}_1$ では平均 4 の手が平均 8 の手よりも悪いと判定された
- $\vec{w}_2$ では平均 4 の手が平均 6 の手よりも悪いと判定された

普通に考えると $\vec{w}_1$ にペナルティを多く加算するべきだと考える．しかし，標準偏差を調べたところ次のようであったとしたらどうだろうか．

- $\vec{w}_1$ では平均 4，標準偏差 2 の手が平均 8，標準偏差 6 の手よりも悪いと判定された
- $\vec{w}_2$ では平均 4，標準偏差 2 の手が平均 6，標準偏差 0 の手よりも悪いと判定された

こうなると $\vec{w}_2$ にペナルティを多く加算するべきだと考えるだろう．つまり，ペナルティとして加算する値は常に 1 としていた場合（以降 01 ペナルティ）と異なり，ペナルティとして加算する値を場合に応じて変化させる場合は，平均的帰結の結果だけではなく平均的帰結の標準偏差も必要となる．

9.2 Welch 検定

平均的帰結と標準偏差から加算するペナルティを求める方法は様々な方法が考えられるが，本章では Welch 検定を用いて求める．

Welch 検定とは t 検定の一種であり，「2 つの母集団の平均は等しい」という帰無仮説を検定するために用いられる．等分散を前提としない検定法であるため，比較対象が等分散でない可能性があるシミュレーション結果を検定に利用することができる．

Welch 検定で使用する検定量 t_0 を式 (9.1)，自由度 ν を式 (9.2) に示す．ここで，$\bar{X}$ は標本 X の平均，n_X は標本 X の標本サイズ，s_X は標本 X の標本分散であり，Y についても同様である．

$$t_0 = \frac{\bar{X} - \bar{Y}}{\sqrt{\frac{s_X^2}{n_X} + \frac{s_Y^2}{n_Y}}} \tag{9.1}$$

$$\nu = \frac{(s_X^2/n_X + s_Y^2/n_Y)^2}{\frac{(s_X^2/n_X)^2}{n_X - 1} + \frac{(s_Y^2/n_Y)^2}{n_Y - 1}} \tag{9.2}$$

標本の平均は平均的帰結 $\vec{x}(s, a, \pi)$，標本サイズとは $\vec{x}(s, a, \pi)$ のシミュレーション回数，標本分散とは $\{\vec{x}_i(s, a, \pi)\}_i$ の標本分散のことである．

Welch 検定で帰無仮説が棄却された場合，それは「2 つの母集団の平均は等しいとは言えない」を意味する．つまり，ペナルティを加算するだけの価値があるということになる．そして，棄却されたときの有意水準が高ければ高いほど，目標の重みでない可能性が高くなっていくことを意味する．本章ではその性質を利用し，最初は高い有意水準から検定を行い，棄却できなかった場合は徐々に有意水準を低くして検定を行う．これにより，適切な有意水準を基にペナルティを設定することができる．本章で使用する有意水準は検定を行う順に 0.1%，0.5%，1%，5%，10%，20%である．ペナルティを計算する関数の処理の流れをアルゴリズム 2 に示す．ここで，$size$ は $\vec{x}$ のシミュレーション回数，σ_x は $\{\vec{x}_i(s, a, \pi)\}_i$ の標本分散，$CalcT$ は t 分布表 [10] から t 値の臨界値を求める関数である．

Algorithm 2 ペナルティ計算アルゴリズム

```
p = {0.1, 0.5, 1, 5, 10, 20}
t_0 = CalcT0(size, σ_{x(s,a,π)}, σ_{x(s,a*,π)}, x(s,a,π), x(s,a*,π))
ν = Calcν(size, σ_{x(s,a,π)}, σ_{x(s,a*,π)})
for i=p.length to 1 do
  t = CalcT(ν, p[i])
  if t_0 > t then
    return 2 × i
  end if
end for
return 0
```

9.3　01 ペナルティとの比較

8.1 節と同じ条件で実験を行った．その結果を 8.1 節と同様に効用関数の重みごとの一致率の推移を図 9.1，設定ごとの一致率の推移を図 9.2 に示し，全てを合計した平均行動一致率の推移を表 9.1 に示す．両図には 8.1 節の結果を点線で示し，本章で示した手法を実線で示した．

グラフからは優劣をつけがたいが，表 9.1 を見ると 1 戦後から 5 戦後までは Welch 検定を用いる方法が有利であることが分かる．最終的な学習結果はさほど変わらないが，学習の高速化には貢献する可能性が高い．

表 9.1: 平均行動一致率の比較

ペナルティを与え方	戦闘回数				
	0.5	1	3	5	8
Welch 検定の検定結果	54.6	67.5	69.4	69.9	70.0
常に 1	58.0	66.7	68.7	69.7	70.3

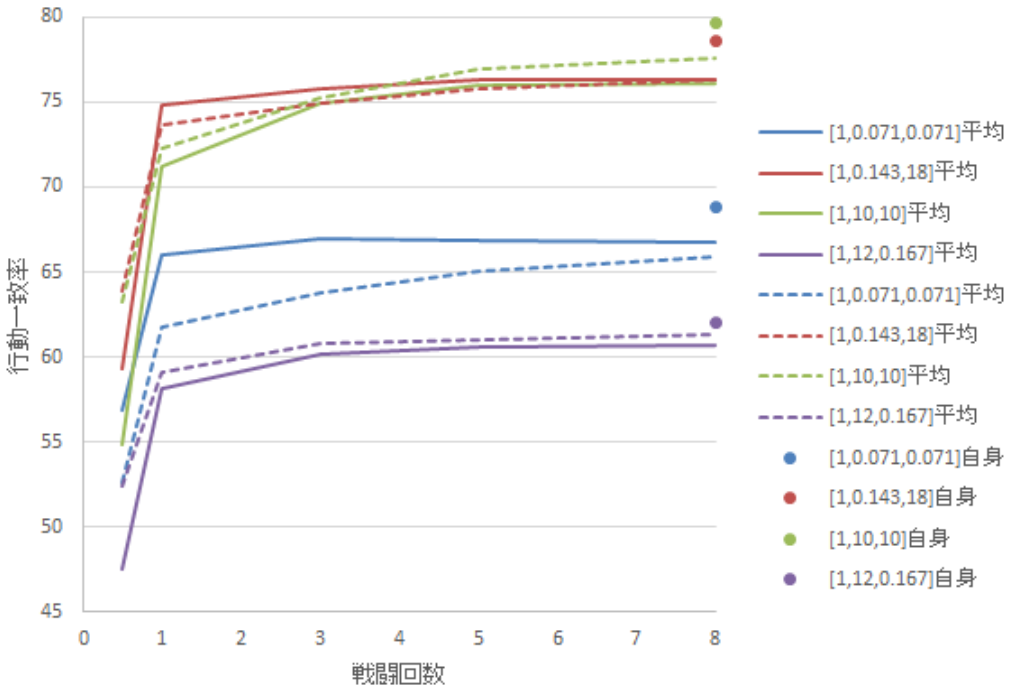

図 9.1: 行動一致率・設定別 点線は 01 ペナルティ

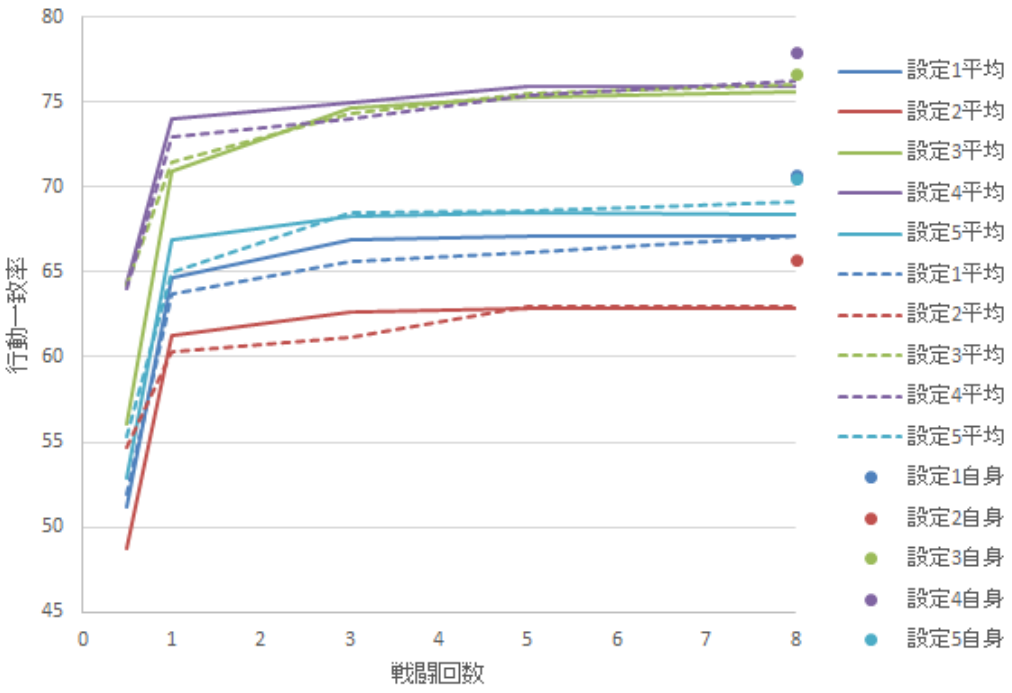

図 9.2: 行動一致率・効用関数の重み別 点線は 01 ペナルティ

第10章　まとめ

本稿では，人間プレイヤが選択した行動から人間プレイヤの効用関数の重みを推定し，それを AI プレイヤの行動選択に活用することでその人間プレイヤにとって満足度が高い AI プレイヤを生成することを目指した．モンテカルロ法におけるシミュレーション戦略を複数用意することで効用要素をより正しく推定することができた．さらに被験者実験により人間プレイヤの効用関数を正しく推定できることを示せた．また，ペナルティの与え方を Welch 検定を使用する方法にした場合，学習が高速化できる可能性があることを示せた．

謝辞

本研究を進めるにあたり，ご指導と助言をいただきました池田心准教授に心より深く感謝いたします．また，Simon Viennot 助教やゲーム情報学研究発表会に投稿した論文の共著者である佐藤直之氏，被験者実験に協力してくださった池田・飯田研究室の皆様に深く感謝いたします．

付 録A　実験に使用した設定

表 A.1: キャラクタのパラメータ　設定1

キャラクタ	HP	MP	攻撃力	守備力	使用可能術技
味方1	134	30	60	28	単体攻撃・小回復・防御
味方2	102	80	44	32	単体攻撃・グループ攻撃 ・小回復・中回復 ・全体回復・防御
敵1〜3	52	0	38	26	単体攻撃
敵4〜6	52	0	38	26	単体攻撃
敵7〜8	52	0	38	26	単体攻撃
敵9〜10	52	0	38	26	単体攻撃

表 A.2: キャラクタのパラメータ　設定3

キャラクタ	HP	MP	攻撃力	守備力	使用可能術技
味方1	138	30	62	30	単体攻撃・小回復・防御
味方2	110	62	46	34	単体攻撃・グループ攻撃 ・小回復・中回復 ・全体回復・防御
敵1〜2	52	0	40	26	単体攻撃
敵3	56	0	56	56	単体攻撃

表 A.3: キャラクタのパラメータ　設定 4

キャラクタ	HP	MP	攻撃力	守備力	使用可能術技
味方 1	142	32	66	36	単体攻撃・小回復・防御
味方 2	112	64	48	38	単体攻撃・グループ攻撃 ・小回復・中回復 ・全体回復・防御
敵 1	84	0	84	20	単体攻撃
敵 2	84	0	44	60	単体攻撃
敵 3	60	0	48	26	単体攻撃

表 A.4: キャラクタのパラメータ　設定 5

キャラクタ	HP	MP	攻撃力	守備力	使用可能術技
味方 1	160	36	74	48	単体攻撃・小回復・防御
味方 2	122	72	52	44	単体攻撃・グループ攻撃 ・小回復・中回復 ・全体回復・防御
敵 1	120	0	54	26	単体攻撃
敵 2	222	0	80	40	単体攻撃・小回復
敵 3	102	32	52	24	単体攻撃

参考文献

[1] Matteo Bernacchia，Hoshino Jun'ichi，AI platform for supporting believable combat in role-playing games, ゲームプログラミングワークショップ 2014 論文集，pp.139-144，2014.

[2] Remi Coulom， Computing Elo ratings of move patterns in the game of Go， International Computer Games Association Journal，30 (2007)，pp.198208，2007.

[3] Sander Bakkes, Pieter Spronck and Eric Postma ，TEAM : The Team-Oriented Evolutionary Adaptability Mechanism, Entertainment Computing - ICEC 2004, pp.273-282, 2004.

[4] 田村坦之，中村豊，藤田廣一， 効用分析の数理と応用， コロナ社，pp.1-28，1997.

[5] 鶴岡慶雅，横山大作，丸山孝志，近山隆， 局面の実現確率に基づくゲーム木探索アルゴリズム ゲームプログラミングワークショップ 2001 論文集，pp.17-24，2001.

[6] 徳田浩，飯田弘之，細江正樹，久保田聡，小谷善行，相手モデルを持つゲーム木探索法についての考察，全国大会公演論文集 第 48 回平成 6 年前期 (2)，pp.123-124，1994.

[7] 藤井叙人，佐藤祐一，若間弘典，風井浩志，片寄晴弘， 生物学的制約の導入によるビデオゲームエージェントの「人間らしい」振舞いの自動獲得， 情報処理学会論文誌 55(7)，pp.1655-1664，2014.

[8] 古居敬大，三輪誠，近山隆，不確定不完全情報展開型多人数ゲームにおける相手モデル化による搾取相手の選択 ゲームプログラミングワークショップ 2011 論文集，pp.46-53，2011.

[9] 保木邦仁， 局面評価の学習を目指した探索結果の最適制御， ゲームプログラミングワークショップ 2006 論文集，pp.78-83，2006.

[10] 本間鶴千代 統計数学入門， 森北出版，p.208，1991

[11] 生井智司，伊藤毅志， 将棋における棋風を感じさせる AI の試作， 情報処理学会研究報告，Vol．2010-GI-24，No.3，pp.1-7，2010.

[12] 吉谷慧，プレイヤの意図や価値観を学習し行動選択するチームプレイ AI の構成，第 29 回ゲーム情報学研究会，pp.1-8，2013.