博 士 論 文

多様な性質のゲームと用途のためのコンピュータプレイヤ拡張の研究

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

佐藤 直之

博 士 論 文

多様な性質のゲームと用途のためのコンピュータプレイヤ拡張の研究

指導教員　池田 心

審査委員主査　池田 心
審査委員　飯田 弘之
審査委員　白井 清昭
審査委員　長谷川 忍
審査委員　西野 順二

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

1420205 佐藤 直之

提出年月: 平成 30 年 3 月

概 要

人間の生活に利益をもたらすために人工知能技術の研究が進められている．研究にあたっては，人間の汎用的な情報処理作業全般を代替できる技術をいきなり実現させることは難しいため，さまざまに分かれた下位分野とともに研究が行われている．その中でゲームの人工プレイヤに関する領域は，現実問題の簡易なモデル・客観的評価のしやすさ，といった点で興味深く，人工知能研究のかなり初期から探究されてきた．その結果としてチェス，将棋，囲碁など様々なゲームで人間のチャンピオンプレイヤを上回る強さの人工プレイヤが開発されるに至り，話題となった．

とはいえ，一方でまだまだ強い人工プレイヤ開発のための手法が十分に整備されないゲームジャンルも多く存在し，そうしたゲームでは個別にはルールベースなルーチンによってある程度対処できるにしても，新しいゲームタイトルが開発されるたびに毎回ルールベースなコーディングを行うことはコストが高い．次から次へと新しいゲームが開発される現代においてそのコストの高さは問題で，人工プレイヤの質の悪化を容易に招いてしまう．そして「単なる強さ」以外にも「人を楽しませるプレイ」や「初級者の教育」または「人間らしい挙動」の実施などに人工プレイヤの目的を変えてみた場合まだまだ技術研究の余地は大きい．これらの目的を持つ人工プレイヤはゲームの面白さに強い影響を持つため商業タイトルでは特に重要と考えられる．

そこで本研究は，ゲーム人工プレイヤの適用可能な「ゲーム種類」と「目的」の拡張を目指した．ゲームと一口にいってもその種類は様々で，また人工プレイヤの目的が変われば必要な技術も大きく変わるため，ゲーム人工プレイヤ研究を４つの下位領域に分割してとらえた．すなわち，ターン制ゲームで強いプレイヤ作成のための研究，ターン制ゲームで強さ以外を目的としたプレイヤ作成のための研究，リアルタイム制ゲームで強いプレイヤ作成のための研究，リアルタイム制ゲームで強さ以外を目的としたプレイヤ作成のための研究，の４領域である．そしてこのそれぞれで対象ゲームの普及の度合い，現代の技術水準からみた取り組み易さ，の２つの観点から見て取り組む価値の高いと感じた未解決課題１つずつの解決を目指した．

ターン制ゲームでの強い人工プレイヤ作成という課題に着目して，本研究はターン制ストラテジーゲームで３種の前向き枝刈り技法による強い人工プレイヤ作成を試みた．この枝刈りにより，ゲーム中のターンあたりの合法手数の組み合わせ爆発の問題に対処した．その結果，既存のチャンピオンプログラムに 74% の勝率をおさめるプレイヤが開発された．

ターン制ゲームでの強さ以外の目的の人工プレイヤ作成の課題に関しては，本研究はRPG ゲームで味方プレイヤの価値観に迎合する仲間役の人工プレイヤ作成を行った．対象の価値観を効用関数でモデル化して，相手の行動の観察からその具体的な形を推定する．そして推定された関数に基づき自分の行動も決定することで，相手の嗜好に一致し

た行動を人工プレイヤにとらせた．被験者実験によって 5 段階の満足度の評価で平均 3.85 点を得て，他の比較用人工プレイヤよりも平均で 0.46 点分ほど高い評価を得た．

リアルタイム制ゲームで強いプレイヤ作成の実現に向けて，本研究は格闘ゲームという対象ゲームに着目した．計算時間が短く限られ，相手の行動に合わせて最適な戦略が変わっていくこのゲームにおいてルールベースな既存人工プレイヤを複数個用意し，相手の挙動に合わせながら切り替えて使い分けることで強いプレイヤの作成を狙った．結果として，既存ルールベース型プレイヤに対し，提案プレイヤが平均で+888 点分（最良で+4500 点，最悪で-4500 点となる状況下）のゲームスコアを搾取した．

リアルタイム制ゲームでの強さ以外の目的のプレイヤ作成という課題に関しては，シューティングゲームで人間らしい挙動の人工プレイヤ開発を試みた．障害物の回避の仕方に細かな挙動の違和感が出やすいこのゲームにおいて，キー操作切り替えの大まかな連続性や，障害物の危険エリアをポテンシャルのマップとして放射状に見積もることによって，なるべく人間らしい挙動が実現されるような経路の探索手法を提案し，評価した．被験者実験により 5 段階評価中で平均 3.1 点を獲得し，本物の人間のスコア 4.2 点には届かないがベースライン人工プレイヤの 2.1 点よりも+1.0 点分だけ良い結果が確認された．

これらの結果の総合として，各領域間を横断するような知見は得られなかったもののそれぞれの領域内で有用な手法を提案することによりゲーム人工プレイヤの適用可能な「ゲーム種類」と「目的」の範囲を拡張し，また多くの人に遊ばれているゲームにおける人工プレイヤの改善にも貢献できたと考える．

目 次

第1章　はじめに

人工知能技術は，人間の知的な処理を計算機に代替させることで社会への広い貢献が期待されている．現時点でも郵便番号の自動読み取りや自然言語翻訳などが人々の生活の役に立っている．万物に対する人間の知的処理を代替するような「万能な人工知能」をいきなり作り出すのは現時点では困難なので，人工知能技術はさまざまな下位分野を持つ．その分野間の横断性を強く意識した研究や技術開発も試みられる一方で，それなりに適用対象やアプローチを絞った上での研究活動も盛んである．その場合の適用対象として分野ごとに着目されるものとしては例えば，ロボットや自然言語処理，ゲームのコンピュータプレイヤ（以降，人工プレイヤと呼ぶ）などが有名である．

本研究ではゲームを適用対象にする．ゲームを人工知能技術の適用対象として見るときには 3 つの利点が考えられる．まずゲームのルールやシステムは複雑な現実世界よりは単純でありながら意思決定は難しい．そのため高度な意思決定の技術を追求する際に，現実の簡単なモデルとしてのテストベッドの役割をゲームは果たせる．次にゲームはしばしば，適用技術に対して解りやすくて客観的な指標を提供する．つまり，チェスや将棋といったゲームの人工プレイヤに適用する人工知能技術には,「ゲーム対戦による勝敗」という解りやすい評価指標がある．最後にゲームはそれ自体人間の文化の 1 つであり，ゲームの優れた人工プレイヤ開発には「人間を楽しませることができる」という社会的な価値がある．そこで本研究はゲームを対象とした人工知能技術の研究を行い,「人間をゲームにおいて楽しませる技術の発展」を主に目指す．

初期のゲーム人工プレイヤ研究はチェスに関するものが有名である．その結果チェスの Deep Blue [1] のように人間のプロプレイヤと互角以上に戦うほどの性能を持つ人工プレイヤが開発された．他のゲームも研究の対象となり，同様の古典的なボードゲームでは将棋の bonanza [2]，囲碁の Alpha Go [3] といった人工プレイヤが同様に人間のプロに匹敵する性能を示した．

これらのゲーム研究で開拓された技術は市販のゲームソフトウェアにも応用され，ゲームを遊ぶ人々の娯楽に直接還元されている．特に古典的ボードゲームを対象として，競技の強さを目的とした人工プレイヤは人間にとって既に十分満足のいく水準に達していることがほとんどである．一方でゲームは全体として種類が多様であり，また人工プレイヤに求められる目的も単純な強さのみとは限らないため，人間の満足のためにはまだまだ改良の余地が大きいような「ゲームジャンル」と「人工プレイヤの目的」の組み合わせも多い．例えば将棋（ジャンル）で人間初級者を上手に指導してあげる（目的）ような人工プレイヤは，目的の難しさにより十分な達成がされていない．また，リアルタイムストラテ

ジーゲーム（ジャンル）で人間上級者より強い（目的）人工プレイヤの実現に関しては，ゲームジャンル側の難しさにより十分に達成されていない．

特に現状では，沢山の人間に遊ばれる人気ゲームジャンルであっても人工プレイヤの強さや遊びの快適さに不満が残ることも多い．更にそうした人工プレイヤの性能向上に関する技術が，現状の既存研究の水準から考えて十分実現可能なように見えるにも関わらず，単純に過ぎる初歩的なハンドコードや，タイトル毎に設計コストが膨大となる複雑なルールベースコーディングにより人工プレイヤが制御されていることもある．このような事態は，そのゲームジャンルが研究対象としてあまり着目されてこなかったり，既存研究は行われたものの実装手順の複雑さ等の理由によって実際の商用ゲームには応用されなかったりしたために生じると想定できる．

よって本研究では,「人気が高いゲームジャンル」そして「需要の大きい目的」の中で，現在の技術で達成の見込みが十分にあるものを実現したい．特に本研究は比較的単純な手法による解決を提案することで商用ゲームへの応用のされやすさを重視したい．そしてそのようなの課題の中で，学際性を考慮し，問題クラスが重複しないような課題群を見出して対処する．その２つの基準を考慮した結果として我々は以下４つの具体的課題を見出した．

1. **ターン制ストラテジーの強いプレイヤ**

   ターン制ストラテジーは累計売上3500万本以上の『civilization』『大戦略』シリーズなどを含む人気のゲームジャンルである．このゲームでは合法手数の多さと駒の複雑な相性関係によりまだ十分強い人工プレイヤが開発されていない問題がある [4]．具体的なリサーチクエスチョンを「組合せによる合法手数が爆発的に増えるゲームで強い木探索プレイヤをどのように開発するのか」に定め，解決に取り組む．

2. **ロールプレイングゲーム（RPG）の快適な仲間プレイヤ**

   RPG は累計売上 1700 万本以上で日本で社会現象にもなった『ドラゴンクエスト』シリーズを含む人気ジャンルである．RPG の仲間キャラクタはしばしば人工プレイヤが操作するが，人間の目指す目的と反した行動をとって不満感を生じうる問題がある．ここでは「人間プレイヤの，個人ごとに様々に異なるゲームスタイルの嗜好をどう読み取って，迎合すれば良いか」をリサーチクエスチョンに定めて解決に取り組む．

3. **格闘ゲームの強いプレイヤ**

   格闘ゲームも累計売上800万本を持つ『ストリートファイター』シリーズを含み，また近年では e-sports としても着目される，活発なジャンルである．格闘ゲームで人工プレイヤは，人間の反射神経よりごく短時間で状況への反応が可能なため単に強いプレイヤを作るだけなら難しくないと予想される．しかしその機械的な短時間の反射による解決は人間プレイヤの不満を招くと予想され，人間と（時間的に）対等な条件下での思考判断力に秀でたプレイヤの開発が待たれるが，その追求は課題と

して未解決である [5]．このゲームではリアルタイム性のため計算時間が短く，また最善戦略が相手の戦略に応じて変化するという難しさがある．この課題では「最善戦略が相手の行動によって変わり続けるリアルタイムゲームで人工プレイヤの強さを向上させるにはどうすれば良いのか」をリサーチクエスチョンとして解決に取り組む．

4. **シューティングの人間らしい挙動のプレイヤ**

   シューティングゲームは特に日本で人気が根強く,『スペースインベーダー』や売上400万本以上の『スターフォックス』シリーズもこのジャンルに分類される．シューティングで人間らしい挙動を行う人工プレイヤはまだほとんど研究の対象になっていない．このゲームではときどき人工プレイヤが人間の対戦相手となり，その際あまりに機械的な動きをとると人間側の面白さが損なわれるリスクがあると考えられる．さらにマップや敵配置などゲームのコンテンツを自動生成するという枠組みにおいても，人間ではなく人工プレイヤがテストプレイを行うことが必要であり，その際には人間らしい人工プレイヤが望ましい．そのため人間らしい挙動の人工プレイヤの実現は興味深い．本研究では，このジャンルでプレイヤの人間らしさ・機械らしさの大きな要素として障害物回避の動作に着目する．この課題については「人工プレイヤがシューティングゲームの障害物を回避する動きを人間らしくするにはどうすれば良いか」をリサーチクエスチョンとして取り組む．

またこの4つの課題は，ゲームジャンルの{ターン制・リアルタイム制}，人工プレイヤの目的の{強さ・それ以外}，の分割によって4種の異なる問題クラスの領域に位置づけられる．ターン制ストラテジーの強いプレイヤ作成は**ターン制ゲーム・強さ目的** の追求の問題であり，シューティングゲームの人間らしい挙動のプレイヤは**リアルタイム制ゲーム・強さ以外の目的** を追求する問題である．

この4種類の領域は，あらゆる「ゲームジャンルと人工プレイヤの目的の組み合わせ」を4分割していて，各クラス内部では，手法にある程度の流用可能性が生じることが期待される．そのため同じ領域内の複数の課題を解決するよりも，本研究のように4つの異なる領域から選んだ課題の解決を試みる方が，より広範な問題群に貢献が波及すると考えられる．更にこの4領域は問題としての性質がバラバラで，それぞれの領域内で有効に働く手法などについての共通性を見出すことは一見難しそうに思われるが，各領域それぞれの課題解決を通じて領域を横断するような有用な知見も発見され得る．そのため本研究はこうした「4領域それぞれからの課題の選択と対処」は有意義であると考える．

このようにして我々は，ゲーム人工プレイヤの様々な「ジャンルと目的」の課題の中から,「**普及の度合いが著しく取り組み易いこと**」および「**異なる4つの問題領域に属していること**」，この2つの要請から4つの課題を見出した．それらの解決を試みることで，今ゲームを遊んでいる多くの人間の娯楽への貢献，そして学際的に幅広いゲームジャンルと人工プレイヤの目的に対応できるような技術の拡張を狙っている．

この論文は以下の形で構成される．第2章でゲーム情報学の全体における人工プレイヤ

研究に関する概観を与える．第３章ではターン制ストラテジーゲームで，可能合法手数の爆発的な増大に対応するための枝刈りつき木探索手法を提案し，性能を評価する．第４章ではRPGゲームで人間プレイヤ個人ごとの嗜好を読み取って迎合する人工プレイヤ手法を提案し，評価のための被験者実験を行う．第５章で格闘ゲームを題材にし，最善戦略が目まぐるしく変わっていくなかで適切な戦略への切り替えを行う人工プレイヤ手法の提案と実験を行う．第６章ではシューティングゲームを題材に，短い計算時間の中で人間らしく余裕をもった回避を人工プレイヤで再現する手法を提案し，実験する．第７章はまとめである．

第2章　ゲームにおける人工知能技術の既存研究

この章ではゲームにおける人工知能技術適用の研究に関して大まかな俯瞰を与えるために，既存研究を取り上げる．各分野内に焦点を絞った，より詳細な研究例の説明については3章から6章までの各章内に専用の項を設ける．

ゲーム対象

ゲームへの人工知能技術適用は古典的なボードゲームから近代的なビデオゲームにまで適用対象を広げてきた．人工知能の概念がコンピュータが普及して間もない頃は，計算機で扱いやすい二人零和有限確定完全情報ゲームとして，ルールが複雑すぎないボードゲームが注目された．特に最初期のゲーム人工プレイヤ研究としてチェスとチェッカーを題材にしたものがとりわけ有名である．チェスはShannon [7] をはじめ様々な研究者が研究に取り組み，1990年代には人間のチャンピオンに勝利をおさめる程の発展をみせた [1] ．チェッカーはSamuelによる人工プレイヤ開発の試み [8] がよく知られ，後の2000年代にゲームが求解される [9] にまで至った．そして他にも単純な部類の二人零和有限確定完全情報ゲームとして例えばNine men's morris [10] や五目並べ [11] が，研究の結果として必勝法を求められた例である．

他のボードゲームについて，オセロでは1990年代にMichaelの人工プレイヤが世界チャンピオンを破った [12] ．将棋では，2015年に情報処理学会が「コンピュータ将棋の実力がトッププロ棋士に追い付いている」との分析を行い，トッププロ棋士に勝つコンピュータ将棋プレイヤの実現プロジェクトの終了を宣言した [13] ．また囲碁は2016年にDeepmind社による人工プレイヤが世界トップクラスのプロ棋士との対局に勝利するほどの成果をおさめた [3] ．さらにGeneral Game Playingという，特定の具体的ゲームに依存しない汎用的なゲームの知的処理を競うための試みがStanford University内のグループからなされて，2005年からその競技会が開かれている [14] ．

それから研究の蓄積に伴い，確定的なゲームだけでなく運の作用があるゲーム，完全情報ゲームだけでなく不完全情報なゲームにおいても活発に研究が進められるようになった．サイコロによる不確定性のあるゲームとして，バックギャモンでは人工プレイヤ [15] が1990年代にTesauroにより開発されて，後に人間の上級者を上回る強さの獲得に成功した．また不完全情報ゲームとして普及の度合いが著しいポーカーにおいては，様々な研究が取り組まれた結果として，2000年代にMartinらがϵ-Nash均衡戦略の導出アルゴリズムを導いて，Heads-up limit Texas Holdemにおいてそのアルゴリズムを利用した人工

プレイヤを実装した [30]．また麻雀では 2015 年に水上らによる人工プレイヤがオンライン麻雀サイトで人間中級者を超えるレートが得たことが報告されている [16]．

近年になってビデオゲームが発達してくると，それまでの現実でのカードやボードを利用した古典的ゲームだけでなくビデオゲームも活発な研究対象として注目されるようになった．ビデオゲームは全体として様々なジャンルに枝分かれしており，そのゲーム的性質も一括りにできないくらい広範であるが，人工プレイヤの競技会が定期的に開催される程の盛り上がりをみせるジャンルも散見される．例えば Star Craft や格闘ゲームは国際会議で定期的に競技会が開かれる [18] [5]．また特定の既存ビデオゲームに限定されない，ビデオゲーム一般用の人工プレイヤ開発を目指す，General Video Game Playing という対象分野 [19] も近年盛り上がりを見せている．

更に，抽象的な計算上の複雑さだけでなく，現実世界の物理的な問題や人間の言語処理を課題として含む方面にも対象は広がっている．実機の機械を制御して競う，サッカーのロボカップ [20] やミニ四駆 AI [21] という分野や,『汝は人狼なりや』という人間の会話をベースにして進行するゲームを行う『人狼知能』といったプロジェクト [22] が進められている．

人工知能技術適用の目的

主にゲームの「強さ」を目的として開発されてきた人工知能技術も，目的に広がりを持つようになってきた．例えばゲームを一緒に遊んで人間を楽しませる目的の人工プレイヤも研究の興味の対象に含まれるようになった．その場合にはよく，対戦相手と強さを同じ程度に揃えることで楽しさをもたらそうと試みられる [23]．しかしそれだけでなく，人工プレイヤの挙動を人間らしいものに近づけることで目的に接近しようとする場合もある [24]．また「人間らしい挙動」の追求に関しては必ずしも人間側の楽しさだけでなく，「人間の思考形態への理解」という別の目的の一環としてなされる場合も想定できる．

さらに，初心者への教育用としての目的を持つゲーム人工知能技術もしばしば提案される．例えば，人間が選択しなかった行動の先に待っていた未来を提示したり [25]，教師役のシステムが良い行動を計算してその部分的な情報を人間に提示する [26] ことで，目的を達成しようとする．

他には，アクションゲームのステージ生成 [27] やパズルの初期配置 [28] など，いわゆるコンテンツの生成を目的とする場合もある．その場合は，生成コンテンツの目的になるべく沿うような技術の適用を試みる．

さらにはゲームの面白さを定量的に評価するための技術適用の例もある．様々なゲームにおいて面白さを定量的に評価するためのゲーム洗練度指標 [29] のモデルに則って，その指標の計算に必要なゲームの情報を人工プレイヤにより自動で集める適用例がある．

適用手法

初期のチェスやチェッカー等の古典的ボードゲームにおける人工プレイヤに関しては，Minimax 型の木探索と $\alpha\beta$ 法が併せてよく用いられた．また，力任せの木探索で読み切れない場合には，末端局面の評価値付けに教師あり学習が導入されて判断の精度を向上させる試み [32] が広く行われた．その教師あり学習のデータ取得の段階において更に木探

索との整合性を考慮した手法 [2] も将棋の人工プレイヤ作成に用いられたり，プロプレイヤーの棋譜から局面の「実現確率」を枝刈りに利用する手法 [50] も提案された．さらにバックギャモンのような確率的な状態遷移のゲームへの対処 [15] として強化学習が適用されたこともあり，機械学習は古来から現在までよく人工プレイヤに利用される．他に確率的なゲームの研究対象としてはポーカーも有名で，これも強化学習のアプローチで Counterfactual regret 指標の最小化でゲームの ϵ ナッシュ均衡導出に成功している [30] ．

1990 年代から広く用いられるようになった手法としてモンテカルロ木探索 [33] は有名である．一般に $\alpha\beta$ 法の木探索は分枝因子が大きなゲームでは読み深さを稼ぐことが難しく，そういう場合は性能が評価関数に大きく支配されてしまう．そして性能が良い状態評価関数の作成には，ゲーム特有の知識や教師あり学習の利用などのアプローチがあるが，囲碁のように良い状態評価関数の作成が困難なゲームもある．対してモンテカルロ木探索は状態の評価をランダムなシミュレーションに頼っており，またゲーム木の探索もシミュレーションの結果に応じて「有望そうな場所」に計算資源を集中させる．

そのような事情もあって囲碁ではモンテカルロ木探索が高い成果をあげ，「特定の致命的な手筋」を見逃しがちな欠点は指摘されているもの [17] ，評価関数作成のコストの低さなどの利点が注目され，他のゲームでも広く使われるようになった．

それからゲームに適用する機械学習といえば，教師データが必要だが高い精度を得やすい教師あり学習と，教師データ不要だが学習にコストがかかりがちな強化学習の 2 つがよく用いられていた．GPU 等の計算機技術の発達に伴い近年は強化学習に対する注目が増した．深層学習を伴わない強化学習も，価値関数の近似ネットワークを訓練するバックギャモンプレイヤ [15] や方策そのものを方策勾配法で学習する将棋プレイヤ [31] など様々な適用例がみられるが，2010 年代に提案された「深層学習を利用した強化学習技法 [34] 」は様々なビデオゲームで高い性能を発揮し着目を集めた．

さてこうした $\alpha\beta$ 法やモンテカルロ木探索などは，商業的なボードゲームタイトルにも適用されていることが予想される一方で，近代的なビデオゲームにおいては商業タイトルでこれらの探索や機械学習の技法が頻繁に利用されてるように見受けられない．もちろん商業タイトルのソフトウェアのコードは公開されていないため，人工プレイヤにどのような技法が使われているかは厳密には不明であるが，if-then 型のルールベースにより動作しているように見えるものが多い．

このような商業と学術間で技術が断絶してるようにみえる状況は問題だと考えられ，それに対する働きかけの例もある．例えば Star Craft II [6] は商業タイトルながら人工プレイヤのルーチンが研究者にも改造可能な形で提供されており，学術研究と実際の商業タイトルにおける適用技術間の距離を埋めている．しかしそのような商業タイトルはごく少数であって，本研究は学研的な方向の知見追求を行うかたわら「多くの人間プレイヤが触れる商業タイトル内の人工プレイヤの質の向上」も狙っている．

第3章　ターン制ストラテジーに$\alpha\beta$法を適用するための３種の枝刈り

本章は会議への投稿論文を元に書き直したものである [35]．本章では，多様な対象と目的のゲーム人工プレイヤのための，ターン制ゲームにおける強い人工プレイヤ作成に着目した研究について述べる．この「ターン制ゲーム・強さ」の領域の中で，幅広い未解決な課題の中から本研究が対象に選ぶのはターン制ストラテジーゲームである．このゲームジャンルは，世間への普及，現在の技術で取り組めそうな難度の水準，という２つの観点から見て，研究対象に相応しいと考える．ターン制ストラテジーの持つ「組み合せによる合法手数の爆発的増大」に対処して強い人工プレイヤを開発するための前向き枝刈り手法を提案する．

3.1　ターン制ストラテジーにおける強い人工プレイヤの意義

強い人工ゲームプレイヤを作ることは人工知能の主要なテーマの１つである．これまでの研究によってチェス [1] や将棋 [41]，囲碁 [3] などの古典的なボードゲームについては人間の上級者に十分匹敵するほど強いプレイヤが作られてきた．その一方でまだ人工プレイヤが人間よりも弱いゲームも存在する．ターン制ストラテジーもそのようなゲームの一種で，人工プレイヤのレベルを人間以上に引き上げるためには更なる研究が必要である．

ターン制ストラテジーではチェスや将棋と同様に，プレイヤが交互に各自の駒（専門的にはユニットという）を操作する．しかしターン制ストラテジーに共通するいくつかのルールによって，強い人工プレイヤの作成は難しい．

例えばターン制ストラテジーでは毎ターン，プレイヤは複数ある駒を全て自由な順序で動かすことができる場合が多い．このルールによってターンあたりのプレイヤの可能な行動選択肢の数は極めて大きくなってしまう．プレイヤが駒を６つしか持っていない場合でさえ，各駒の可能な行動選択肢が 10 個の場合，組合せでターンの可能な行動選択肢数は 7200 億にも達する．他にもターン制ストラテジーでは様々な初期局面が存在することや駒同士が戦いにおいて複雑な相性関係を持つことが人工プレイヤ開発を困難にする．様々な初期局面の存在は教師あり学習の適用を難しくし，相性関係は高精度な局面評価関数の設計を困難にする．

これらの難点はモンテカルロ木探索により対処されることが多く，$\alpha\beta$法などミニマックス探索型の手法はあまり適用されない．しかしながらターン制ストラテジーはゲームの

枠組みとしてチェスや将棋と似ており，それらのゲームでは$\alpha\beta$法が大きな成功を収めている．加えて，人間プレイヤはしばしば可能な行動の数ターンにまたがる応酬を何パターンか先読みすることで次の着手を決定するが，この思考様式は$\alpha\beta$法による先読みと似ている．ただし人間プレイヤは膨大な可能着手の中から有効そうなものに絞り込むことで，十分なターン数にまたがる先読みを行っている．そこで本研究はターン制ストラテジーで，枝刈りによる着手絞り込みを伴う$\alpha\beta$法を適用した．

本研究は$\alpha\beta$法の枝の数を減らすための３つの工夫を用いた．この工夫により可能な探索深さを増やし，人工プレイヤの性能向上を狙う．その３つの工夫を以下に述べる．

- 駒の行動順序の固定
- 駒行動の絞り込み
- 行動可能な駒の数の制限

これらの工夫は前向き枝刈りであるため，優れた着手を見逃すリスクが含まれる．しかし適度な度合いでこれら導入することで人工プレイヤの性能は総合的に向上すると本研究では考えている．特にターン制ストラテジーでは合法手数の多さのため，ナイーブな実装による$\alpha\beta$法だと現実的な時間では次の敵の手番にさえ先読みが及ばない．これを本研究の枝刈りによって「敵の手番の行動を読める」程度に読み深さを延長するだけでも性能に大きな改善が見込めると本研究では考える．

3.2 関連研究

3.2.1 ミニマックス探索と枝刈り手法

ミニマックス探索はチェスや将棋など様々なゲームの人工プレイヤに適用されてきた．しばしば実装者は探索の深さを稼ぐためにミニマックス木の枝のいくつかを削除する．このテクニックは一般に枝刈りと呼ばれ，前向きと後ろ向きの2種類に分類される．後ろ向き枝刈りは探索結果に影響を与えない方法で，$\alpha\beta$法が有名である [69]．

一方で前向き枝刈りは探索結果に影響を与えてしまうリスクを伴いながらゲーム木のいくつかの枝を削除する．例えば1960年代のチェスプログラム用で用いられた手法では，move ordering技法 [43] によって順序づけられた上位n個の枝のみを各ノードから探索するtapered n-best search [42] が前向き枝刈りである．

より近年の手法では，$\alpha\beta$法のα値とβ値を利用した前向き枝刈りを行う場合がある．例えばfutility pruning [44] やnull move pruning [45] は局面評価値の上限と下限を着手から見積もって利用する前向き枝刈り手法である．この見積もった値がノードのα値やβ値を更新する見込みが薄いとき，これらの値はその着手の枝を読みから除外する．また$\alpha\beta$法に与えるα値とβ値の差（いわゆる探索窓）を狭めることで頻繁に$\alpha\beta$法の枝刈りを起

こす工夫として Nega Scout もよく知られる [48]．これらの手法はチェスだけでなく将棋等にも適用され，深さ 10 以上もの探索を可能にしている [47]．

多くの枝刈り手法はチェスの研究から得られたが，将棋とオセロを題材にした研究から提案された手法として ProbeCut [49] と実現確率探索 [50] が有名である．ProbeCut はオセロに適用され，似たような局面群のおよその評価値をオフラインで計算することで，オンライン探索中にノード評価値の上限と下限の見積もりを考えて枝刈りの頻度を増やす手法である．実現確率探索は将棋を適用対象に性能評価が行われ，着手の特徴量を用いて「上級者の対局で選ばれる確率」を棋譜から見積もり，あまり到達の可能性が高くなさそうな局面に対して探索を打ち切る．

3.2.2 合法手が多いゲームでの探索

ターン制ストラテジーは合法手が多いためゲーム木探索を難しくする．他に合法手が多いゲームとしてここでは Amazon と Arimaa を取り上げ，それぞれで行われてきた工夫について述べる．

Amazon は駒を一直線に動かした後に「弓矢」をまた別の一直線上の一マスに向けて射る行動が伴って，その移動と射的の組合せで合法手の数が大きくなるゲームである．探索の工夫としては，移動の評価値と射的の評価値を分離して扱い，評価値が低くなりそうな枝の探索をあらかじめ省略する Selective Search が試みられた [52]．また，探索の枝刈りではなくゲームの評価関数については，駒の支配する領域や移動の自由度に基づく特徴量によって状態評価の精度を上げる試みがなされている [51]．そして，モンテカルロ木探索による強さの向上を狙った研究も存在し，いくつかのゲーム特有の知識に基づいて偏ったシミュレーションにより高い精度の獲得を狙った例もある [54]．

Arimaa は駒を 1 手番に 4 回動かせるゲームであり，駒の間に強弱の相性関係がある点も合わせてターン制ストラテジーと類似する点がそこそこ多い．Null move pruning や move ordering による探索の効率化 [57] [58] や，1 手番に駒を 3 回しか動かさない着手やあきらかな手順前後可能パターンの読み飛ばしによる工夫 [56] が知られる．また機械学習の部分的な適用例も見られ，Move ordering を，人間上級者の棋譜からの学習により高精度化した試み [55] や棋譜の比較学習による評価関数の作成 [53] も行われた．

3.2.3 ターン制ストラテジー

ターン制ストラテジーの研究としては Cid Meier の Civilization シリーズ [59] やそのクローンに関するものが多い [61] [62] [63]．しかしこのシリーズは「駒同士の戦闘」の他に「経済」や「外交」といった複雑な要素が多く，扱いづらいと考える．そこで駒同士の戦闘に焦点を当てたゲームを本研究では対象に選んだ．

この駒同士の戦闘に焦点をあてたタイプの研究としては，まず進化計算技術による人工プレイヤを Advance Wars のクローンに適用したものがある [60]．また UCT 探索手法 [64]

をファジー関数 [65] と組み合わせたプレイヤを，ターン制ストラテジープラットフォームの TUBSTAP [66] で実装した研究もある．本研究もこの TUBSTAP プラットフォームを用いて提案手法の実装と性能評価を行った．

これらの駒の戦闘に関する既存手法はそれぞれなんとなくの盤面の良さを見積もるのに向いているが，$\alpha\beta$ 法ベースの先読みと異なりたった数手分の長さしかなくても重要な読み筋を見逃しやすい．ターン制ストラテジーは概してたった数手で大きく形勢が変わってしまうゲームなため，$\alpha\beta$ 法ベースの先読みを導入することでそうした重要な影響力を持つ筋を見逃さず，強さの向上に大きく寄与すると本研究では考えた．

3.3 TUBSTAP プラットフォーム

TUBSTAP はターン制ストラテジーの公開プラットフォームである．ゲームルールは “Famicon Wars DS2” [67] をモデルにして設計されている．スクリーンショットを図 3.1 に示す．このプラットフォーム人工プレイヤの競技会が定期的に開催されており，その参加者のコードも自由に利用可能である．

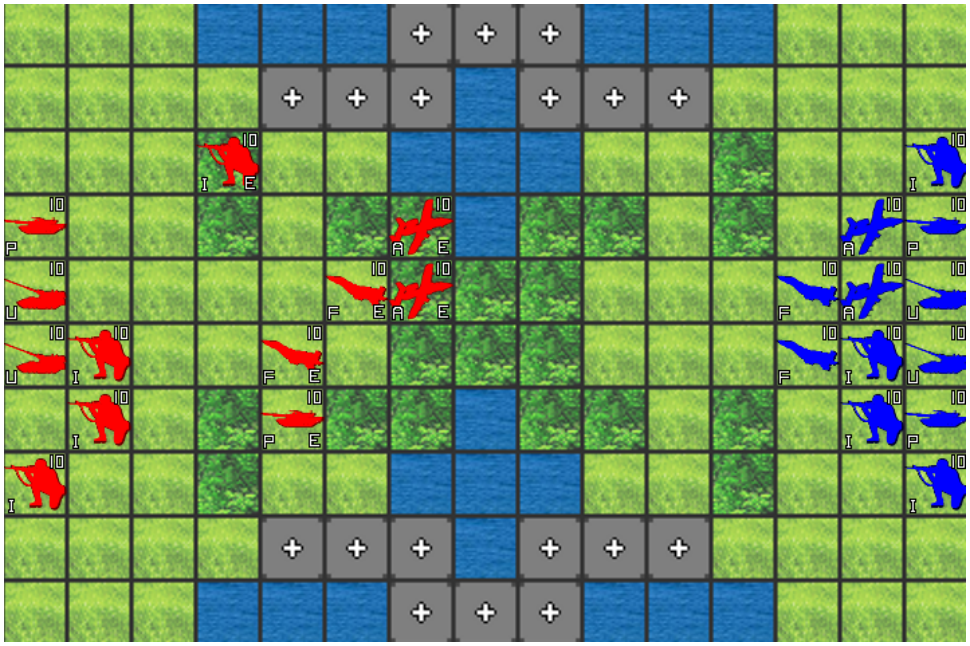

図 3.1: TUBSTAP プラットフォームのスクリーンショット

このプラットフォームでのゲームルールを大まかに述べる．各ターンに各プレイヤは自分の駒を好きな順序で 1 ターン内に全て動かすことができる．それぞれの駒は攻撃，移動，または「移動してから攻撃」を行動としてとることができる．攻撃行動は，攻撃を受けた相手の駒の HP（耐久度）を減らす．HP がゼロになったときその駒はゲームから取り除かれる．そしてすべての駒を失ったプレイヤはゲームに負ける，というルールになっている．

このプラットフォームでは6種類の駒が存在する．戦闘機，攻撃機，戦車，迫撃砲，対空戦車，そして歩兵でありそれぞれ記号でF，A，P，U，R，Iと略記する．さらにゲーム盤のマスは5種類の地形マスから成り立っており，各マスは山，森，平原，道路，海のいずれかの地形マスである．それぞれの駒種類と地形マスに関する数値を表3.1と3.2に示す.「攻撃力」と「防衛効果」は攻撃のダメージ（攻撃による相手駒のHPの減少値）を決定づける数値である．式3.1にその具体的な決定式を示す.

表 3.1: 駒の攻撃力

Defense / Attack	F	A	P	U	R	I
F	55	65	0	0	0	0
A	0	0	85	115	105	105
P	0	0	55	70	75	75
U	0	0	60	75	65	90
R	70	70	15	50	45	105
I	0	0	5	10	3	55

表 3.2: 地形マスの地形効果と移動コスト

地形	山	森	平野	道路	海
	[地形効果]				
A, F	0	0	0	0	0
R, I, P, U	0.4	0.3	0.1	0	0
	[移動コスト]				
A, F	1	1	1	1	1
P, U, R	∞	2	1	1	∞
I	2	1	1	1	∞

$$\text{ダメージ} = \frac{(\text{攻撃力}) \times (\text{攻撃駒 HP}) + 70}{100 + (\text{地形効果}) \times (\text{防御駒 HP})} \tag{3.1}$$

例えばHP7の駒種Aの駒が敵のHP9の駒種Pを攻撃した場合，敵のPが受けるダメージは5となる.

駒が移動行動で1ターンに到達可能なマスは，駒の「移動力」および道のり上に存在するマスの「移動コスト」合計で決定される．駒種ごとの「移動力」はF，A，P，U，R，Iでそれぞれ9，8，6，6，6，3である．各地形マスの「移動コスト」を表3.2に示す．ある駒の移動コストがあるマスに1ターンで到達可能かは，その移動経路上のマスの移動コスト合計が駒の移動力以下か否かで決まる．駒は移動時に，敵の駒がいるマスを通ることはできないが味方の駒がいるマスは通り抜けることができる（ただし同じマスに留まることはできない）.

Uを除く駒種は移動した後に隣接するマス上にいる敵駒1つを攻撃することができる．攻撃を行うと同時に敵駒からの「反撃」として，敵駒からその駒への攻撃が自動的に行われる．駒種Uは離れたマスにいる敵の駒にのみ攻撃ができる．2または3マス分離れた敵の駒に攻撃ができ,「反撃」は行われない．しかし駒種Uは移動と攻撃を1ターン内に両方行うことはできない.

このプラットフォームのルールは多くのターン制ストラテジーに共通して見られる主要ルール要素を網羅している一方で，既存タイトルよりも簡単に設計されている．よって，このプラットフォームで強い人工プレイヤ作成に成功しても他ゲームでの強い人工プレイヤ作成にとって十分ではないものの，取り組み易さを理由に本研究の手頃なテストベッドに選んだ.

3.4 接近法

本研究が適用した3つの工夫ここで述べる．本研究では木探索の計算コストを減らすためにいくらかの枝を枝刈りする方法を考えた．これらの工夫は，単に計算時間を減らすだけでなく，その結果より深い先読みの探索が可能になる点で重要である．

3.4.1 駒の行動順序の固定

ターン制ストラテジーでは1ターン中に「どのような順序で駒を動かすか」がときどき重要になる．とはいえ，駒の動く順序が全く結果に影響を与えない場面はかなり多い．加えて，駒の動く順序が重要になる状況の中でもしばしば，たった数個の駒たちの行動順序のみが問題になる（例えば，沢山ある自分の駒の中で駒Xが駒Yよりも先に行動するか後に行動するかのみが結果に影響を与える，という状況も多い）．人間プレイヤの多くも着手を読むときに，しばしば多くの局面で駒の順序を無視して先読みをする．ゆえに本研究では，各プレイヤが駒を動かす順序を複数（または単一）のパターンのみに限ることで計算量を減らすことを試みた．

今回本研究が考慮した駒の行動順序パターンを以下に述べる．ただし各プレイヤの手持ちの駒全てに1からNまでの識別番号が振られているとする．

- **Forward** $1, 2, 3, \ldots, N$.
- **Backward** $N, (N-1), (N-2), \ldots, 1$.
- **Cut-Forward** $(\frac{N}{2}+1), (\frac{N}{2}+2), \ldots, N, 1, 2, 3, \ldots, \frac{N}{2}$．Forwardの前半と後半の入れ替え．
- **Cut-Backward** $(\frac{N}{2}-1), (\frac{N}{2}-2), \ldots, 1, N, (N-1), \ldots, \frac{N}{2}$．Backwardの前半と後半の入れ替え．

パラメータの設定により本研究の提案人工プレイヤは，このうち1つまたは複数のみを駒の行動順序として考慮し，それ以外の行動は探索木より枝刈りする．

3.4.2 駒行動の絞り込み

ここで使われる専門用語に関して整理する．厳密にいってターン制ストラテジーでは駒に，攻撃行動と移動行動と「移動してから攻撃」の行動がある．しかし簡便のためこの「移動してから攻撃」の行動も「攻撃行動」という用語に含める．さて本研究では各駒の攻撃行動と移動行動に異なる種類の絞り込みを適用する．

[移動行動の絞り込み]

駒ごとに可能な移動行動の数はえてして大きい．そのため本研究では移動行動を，敵の駒からの攻撃射程に基づいてグループ分けする．このグループ分けの手続きを図3.2に示

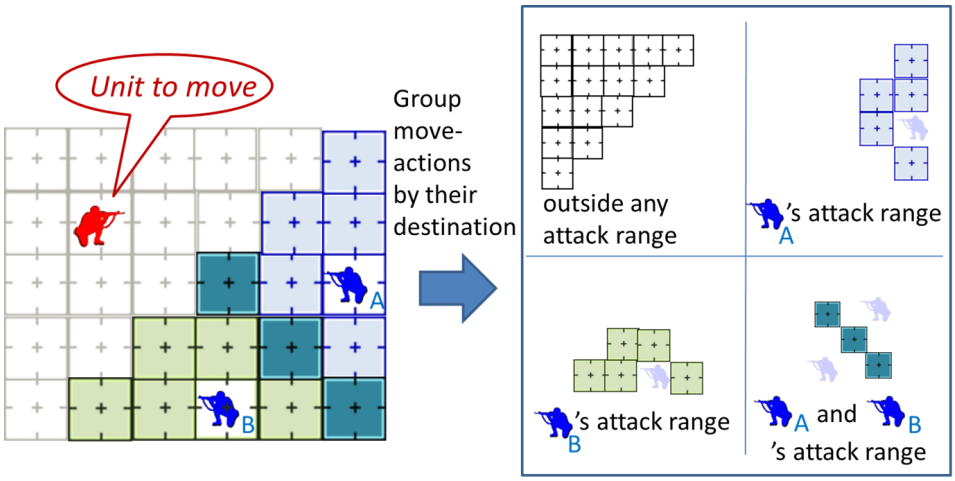

図 3.2: 移動行動のグループ分け．各グループから 1 つの移動行動のみ生成される．

す．そして本研究では各グループから 1 つずつ移動行動を選び，それ以外を枝刈りする．このグループ分けは，各グループ内で「どの敵駒から次ターンに攻撃され得るか」が共通しているため，その中からの 1 つのみを考慮した先読みは「重要な手筋の読み落としのリスク」がある程度低く計算量を減らせると本研究では考える．

さて移動行動をグループ分けした後，その中から行動を 1 つだけ選ぶ方法はいろいろあり得る．だが本研究ではひとまず，以下の優先順位に基づき移動行動 1 つの選択を行う（1. に示した項目の方が高く優先される）．

1. 移動先の地形マスから受ける地形効果の高い順

2. 盤上に配置された全ての駒の中心座標からのマンハッタン距離の近い順

この選択法により複数の移動行動の候補が残った場合，ランダムに 1 つのみを選択する．

[攻撃行動の絞り込み]

移動行動のときと同様のグループ分けを，攻撃行動についても行う．1 つの駒の攻撃行動を，その攻撃対象の駒ごとにグループ分けする（ある駒から別の駒 1 つに対する攻撃行動は，場合により 2 つ以上ありえることに注意されたい）．その上で本研究では各グループから 1 つだけの攻撃行動を除いて他の攻撃行動全てを枝刈りする．その「1 つだけの攻撃行動」を複数の行動から選び出す基準は，次の敵ターンになるべく少ない数の敵の駒から攻撃を受け得るものを優先して選ぶ．複数の候補が残った場合はランダムに 1 つを選び出す．

3.4.3 行動可能な駒の数の制限

詳細な説明に移る前に，混乱を避けるためこの節に登場する専門的な用語の定義を以下に与える．

- **駒行動**：1 つの駒による単位的な行動．移動行動または攻撃行動である．
- **プレイヤ行動**：プレイヤが 1 ターン内に行う，複数の駒による駒行動．もしプレイヤが N 個の駒を持っているとき，そのプレイヤ行動は最大 N 個の駒行動から成る．

ターン制ストラテジーでは可能なプレイヤ行動の数は駒の数に対して指数的に増大する．もしもプレイヤが N 個の駒を持っており，それぞれの駒が M 個の駒行動が可能だとして，ミニマックス探索木で局面の先読みをする場合を考える．このとき，たった 1 ターン分のプレイヤ行動全てを先読み（つまり深さ 1 の探索）するだけで $N!M^N$ 個の葉ノードが必要となる．この値は，通常規模のゲーム局面を想定した場合ですら極めて大きな値になり得て，$M = 30$ かつ $N = 6$ ぐらいの場合でも 5200 億個の葉ノードが深さ 1 の探索に必要である．

そのため本研究では 1 度の探索で先読みに含まれる駒の数を減らす．その手続きは図 3.3 に示される．プレイヤは手持ちの駒全てが行動し終えるまで以下 2 つの手続きを交互に繰り返す．

- ある限られた数の駒のみが可能行動を生成する木探索により最良のプレイヤ行動を見つけ出す．
- そのプレイヤ行動に含まれる最初の駒行動のみを実行に移す．

この工夫によって計算量の指数的な増大をある程度抑制できる．各プレイヤが N 駒を所持しそれぞれの駒が M 個の駒行動が可能であるとき，単純に考えれば，深さ D の木探索を行うのに必要なノード数は $(N!M^N)^D$ である．しかしこの駒数の制限の工夫を適用すれば，N' 個 $(< N)$ の駒のみが行動可能な木探索を高々 N 回適用することで $N(N'!M^{N'})^D$ 以下に必要なノード数が減る．つまり必要な探索ノード数が $\frac{1}{N}(\frac{N!}{N'!}M(N-N'))^D$ 分の 1 になる．

3.5 予備実験 1：駒の行動順序の固定

まず本研究では「駒の行動順序の固定」による枝刈りが人工プレイヤの性能に与える影響を確かめるための実験を行った．$\alpha\beta$ 法による人工プレイヤを用意し，駒の行動順序固定による枝刈りを適用した．以降このように実験のために用意した人工プレイヤを「提案プレイヤ」と呼ぶ．この実験の提案プレイヤは 2 種類のパラメータを持つ．1 つはゲーム木の探索深さであり，もう 1 つは駒の行動順序パターン数である．そしてプレイヤの性能は適当な相手役プレイヤとの対戦によって測られる．

3.5.1 提案プレイヤの設定

提案プレイヤの探索深さは 1 または 2 である（駒行動ではなく，先読みする “プレイヤ行動” の数を深さとする）．ここで考慮される駒の行動順序は {**Forward**}（3.4.1 節参照），

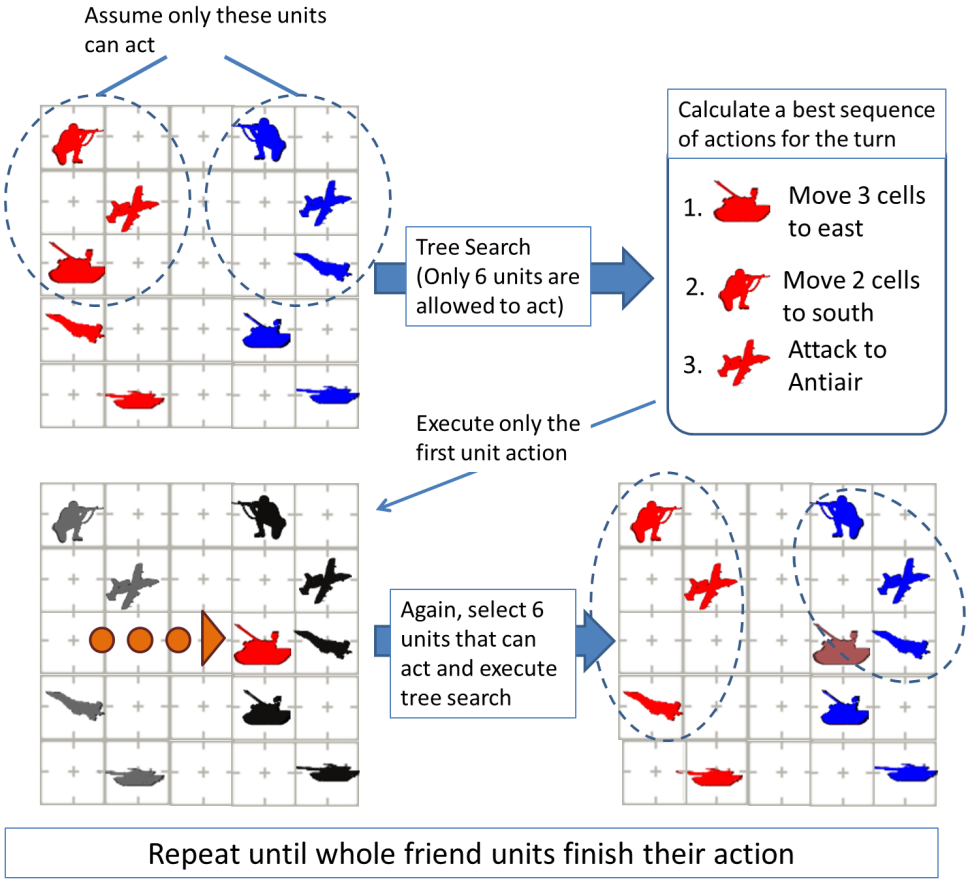

図 3.3: 制限された駒数による木探索．深い 1 回限りの木探索の代わりに，限られた数の駒しか動かないような浅い木探索を何度も繰り返すことでそのターンのプレイヤ行動を決める．

{**Forward**，**Backward**}，{**Forward**，**Backward**，**Cut-Forward**，**Cut-Backward**}，または，あらゆる全ての行動順序を考慮する，のいずれかである．木探索に適用される評価関数については詳細を付録に示す．

3.5.2　実験設定

提案プレイヤはナイーブな UCT 探索プレイヤ [64] を相手に対戦する．この相手役 UCT プレイヤは 10,000 回のプレイアウトごとに駒行動を 1 つ生成する．図 3.4 から 3.6 に示すマップが対戦に使われ，それぞれで 200 戦ずつの対戦が行われた．提案プレイヤがそのうち 100 回で先手番，残り 100 回で後手番である．引き分けの試合については両プレイヤにとっての 0.5 勝分として扱った．

図 3.4: マップ X．赤（F6, A7, R7, P7），青（F10, A10, R10, P10）．先手は赤

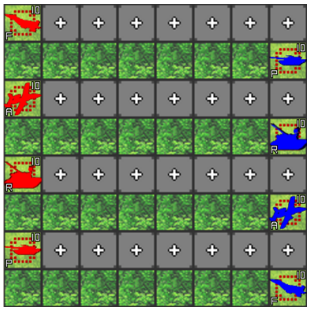

図 3.5: マップ Y．赤（F10, A10, R10, P10），青（F10, A10, R10, P10）．先手は赤

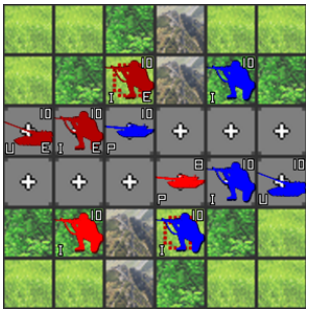

図 3.6: マップ Z．赤（F10, A10, R10, P10），青（F10, A10, R10, P10）．先手は赤

3.5.3 結果

この対戦実験の結果を図 3.7 と 3.8 に示す．考慮する駒の行動順序パターン数が増えるほど性能は高くなる傾向がある．同時に，計算時間もこのパターン数の増加に伴い増加している．考慮する順序パターン数を減らすことによる勝率の下降は高々10% だが計算時間は場合によって 10 分の 1 以下にも減少している．この結果から本研究では，この枝刈り手法は人工プレイヤの性能を効果的に向上させることができると考える．

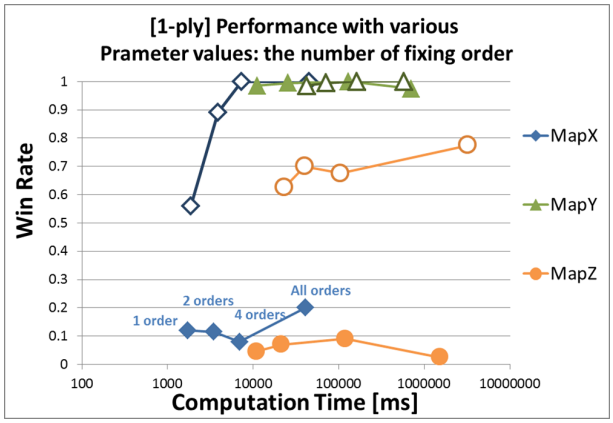

図 3.7: UCT プレイヤーへの勝率．深さ 1．考慮する行動順序パターンが変化．マーカーの塗りつぶしと中抜きはそれぞれ先手番と後手番時のデータ．

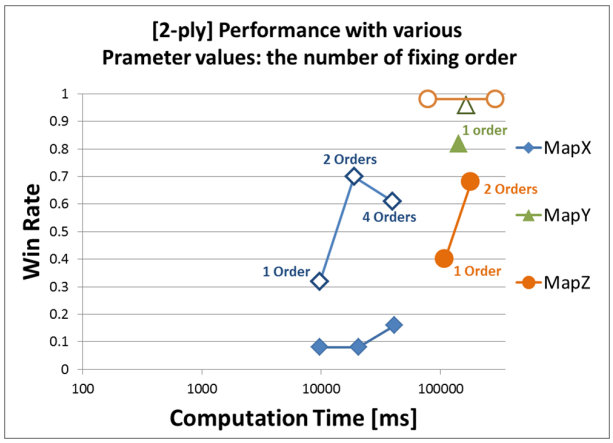

図 3.8: UCT プレイヤーへの勝率．深さ 2．考慮する行動順序パターンが変化．マーカーの塗りつぶしと中抜きはそれぞれ先手番と後手番時のデータ．1 手の計算時間が 300 秒を超えたプロットは除外．

しかしこの手法による，ある種の欠陥も想定できる．ゲームにおけるある特定の状況

で，敵の駒全てを “狭い道” の上で除去する必要があるときこの提案手法は最善手を高確率で見落とすことが予想される．この種の局面ではよく可能な駒の行動順序のうち，ある特定の少数のものだけが敵の駒全ての除去に成功するためである．

3.6 予備実験 2：駒行動の絞り込み

「駒行動の絞り込み」による枝刈りの適用が性能に与える影響を実験により確かめる．

3.6.1 提案プレイヤの設定

本研究では提案プレイヤとして，駒行動の絞り込みを適用した $\alpha\beta$ 法プレイヤとそうでない $\alpha\beta$ 法プレイヤを用意した．局面評価関数は 3.5 節と同様だが，3.5 節の実験と異なる点としてこの実験で提案プレイヤは全ての可能な駒の行動順序を木探索で考慮する．本研究では「移動行動のみ絞り込み」,「攻撃行動のみ絞り込み」,「移動行動と攻撃行動の絞り込み」を適用した場合の性能と計算時間の変化を観察した．

3.6.2 実験設定

ほとんどの設定は 3.5 節の実験と同様である．$\alpha\beta$ 法の提案プレイヤが 10,000 プレイアウトの UCT プレイヤと図 3.4 から 3.6 のマップで対戦する．提案プレイヤの探索深さは 1 または 2 で，枝刈りのオプションは以下のいずれかである．

- **Both:**移動行動と攻撃行動の両方に絞り込みを適用する．
- **Move:**移動行動のみ絞り込みを適用する．
- **Attack:**攻撃行動のみ絞り込みを適用する．
- **No-Prune:**移動と攻撃行動の両方に絞り込みを適用しない．可能な合法手全てを読みに加える．

3.6.3 結果

実験結果を図 3.9 と 3.10 に示す．攻撃行動の絞り込みは，わずかな差であるが性能を劣化させているようにみえる．その一方で移動行動の絞り込みは性能をあまり劣化させず，なおかつ計算時間をしばしば 10 分の 1 以下に抑えている．よってこの設定においては，駒行動の絞り込みは大きな性能の劣化を招かずに計算時間の節約に成功した．

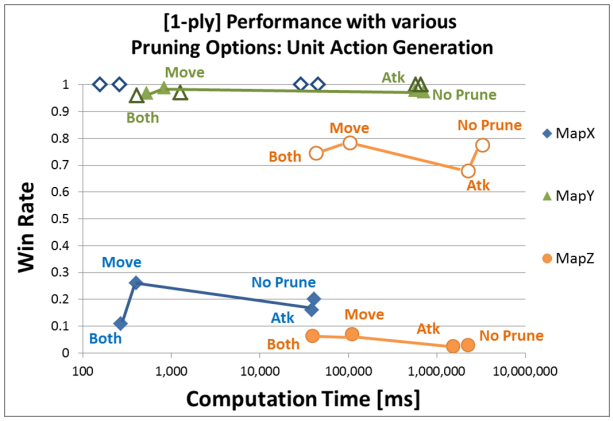

図 3.9: UCT プレイヤーへの勝率．深さ 1．行動絞り込みの種類が変化．マーカーの塗りつぶしと中抜きはそれぞれ先手番と後手番時のデータ．

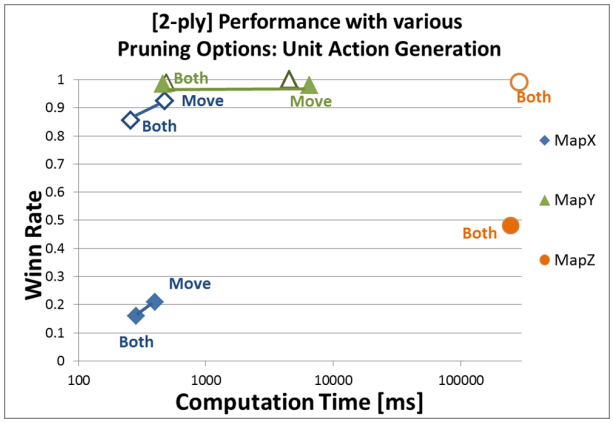

図 3.10: UCT プレイヤーへの勝率．深さ 2．行動絞り込みの種類が変化．マーカーの塗りつぶしと中抜きはそれぞれ先手番と後手番時のデータ．1 手の計算時間が 300 秒を超えたプロットは除外．

3.7 予備実験 3：行動可能な駒の数の制限

予備実験の最後として，行動可能な駒数を制限する工夫が木探索に及ぼす影響を確かめた．

3.7.1 提案プレイヤの設定

3.5 節の実験と同様に $\alpha\beta$ 法のプレイヤを用意し，行動可能な駒数を制限した上で深さ 1 または 2 の木探索を複数回行うことでプレイヤ行動を決定させた．

3.7.2 実験設定

一度の木探索につき行動可能な｛味方の駒の数，敵の駒の数｝のパラメータを｛1, 1｝，｛3, 3｝，｛4, 4｝として対戦実験を行った．対戦相手は 10,000 回のプレイアウトで行動を決定する UCT 探索プレイヤである．マップは 3.5 節までとは異なって，TUBSTAP ver 1.07 に付随するサンプルマップを使用した．マップを変えた理由は，この枝刈りは他 2 つよりも計算時間の減少の度合いが著しいため，3.5 節までで使用したマップより駒数の規模の大きな（それゆえ実際のゲームで使われるサイズにより近い）マップでの人工プレイヤの動作が可能になるためである．対戦は各マップ 200 戦ずつ行った．

3.7.3 結果

実験結果を図 3.11 から図 3.11 に示す．パラメータの値が大きくなるほど勝率と計算時間が増加する．加えて，探索の深さを大きくするほどだいたいの場合性能が改善した．よってこの枝刈り手法の適用によって，ある程度の性能劣化は起こるものの計算時間が減少することが確認できた．

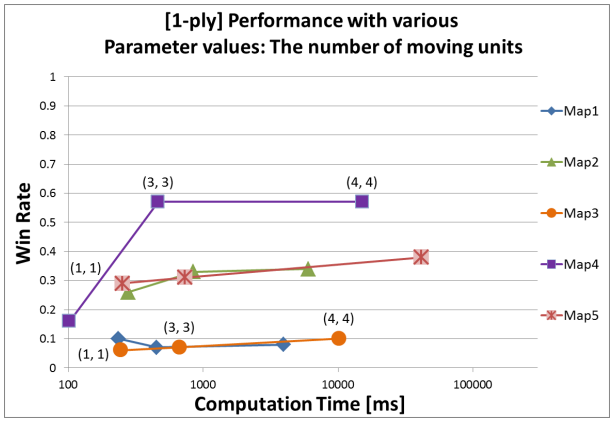

図 3.11: UCT プレイヤーへの勝率．深さ 1．パラメータ（M, N）は “M 個の味方駒と N 個の敵駒のみが行動可能な木探索による着手決定” の意．

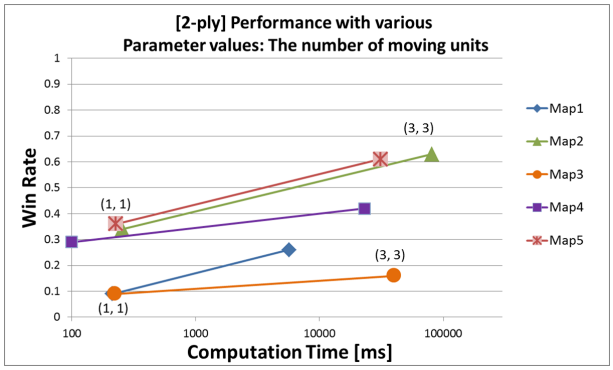

図 3.12: UCT プレイヤーへの勝率．深さ 2．パラメータ（M, N）は “M 個の味方駒と N 個の敵駒のみが行動可能な木探索による着手決定” の意．1 手の計算時間が 300 秒を超えたプロットは除外．

3.8 総合的な性能評価実験

これまでに示した 3 種類の枝刈り手法全てを適用した場合の性能評価を行う．提案プレイヤはこれまでに示した枝刈りの工夫全てを用い，敵役のプレイヤとして TUBSTAP の 2016 年 AI 競技会の優勝・準優勝プレイヤ [68] と対戦を行う．

3.8.1 提案プレイヤの設定

この実験の提案プレイヤは $\alpha\beta$ 法をベースとして以下のオプションおよびパラメータ設定を持つ．

- 探索深さ 2
- 駒の行動順序パターンを 2 通りのみ考慮
- 木探索の中で，味方の駒の攻撃行動と移動行動を絞り込み

- 木探索の中で，敵の駒の攻撃行動を絞り込み（かつ移動行動は一切生成しない）
- 一度の木探索につき行動可能な｛味方の駒の数，敵の駒の数｝は｛5，10｝

3.8.2 実験設定

対戦の設定を以下にまとめる．

- 2016 年の TUBSTAP の AI 競技会 [68] で使用された 5 種のマップを使用
- 同競技会で 1 位と 2 位のプレイヤ “M-UCT” と “DLMC-PW55” を相手に対戦
- 各マップと対戦相手ごとに 200 戦ずつ試行．うち 100 戦のみ提案プレイヤが先手番
- 各プレイヤの計算時間は毎ターン 10 秒程度になるようパラメータ設定

また本研究では 3 種の枝刈りを一切適用しない $\alpha\beta$ 法プレイヤも用意したが対戦実験には加えなかった．そのプレイヤはこれらのマップ上では計算時間を 60 秒費やしても，深さ 1 の先読みも終了しなかったためである．

3.8.3 結果

実験の結果を表 3.3 に示す．提案プレイヤは 5 つ中 4 つのマップにおいて，競技会の 1 位 2 位のプレイヤに大きく勝ち越した．さらに全てのマップを通しての総合的な平均勝率は 70% を上回った．このように高い勝率を提案プレイヤが示した理由は，本研究の枝刈り手法が「強い手の読み落とし」のリスクが十分低いまま「木探索の深さを延長する」ことに成功したためだと考える．

表 3.3: Win rates against 1st-rank and 2nd-rank AI players (200 matches for each map)

1 位プレイヤ “M-UCT” に対する勝率 (および 95%信頼区間)						
	Map-Fujiki	Map-Ishitobi	Map-Muto	Map-Sato	Map-Takahashi	平均
先手	90 (±5.9)	75 (±8.5)	41 (±9.6)	93 (±5.0)	66 (±9.3)	73 (±3.9)
後手	94 (±4.7)	65 (±9.3)	26 (±8.6)	98 (±2.7)	92 (±5.3)	75 (±3.8)
計	92 (±3.8)	70 (±6.4)	34 (±4.2)	96 (±1.7)	79 (±3.6)	74 (±2.7)
2 位プレイヤ “DLMC-PW55” に対する勝率 (および 95%信頼区間)						
	Map-Fujiki	Map-Ishitobi	Map-Muto	Map-Sato	Map-Takahashi	平均
先手	89 (±6.1)	65 (±9.3)	82 (±7.5)	97 (±3.3)	63 (±9.5)	79 (±3.6)
後手	82 (±7.5)	53 (±9.8)	78 (±8.1)	97 (±3.3)	80 (±7.8)	77 (±3.7)
計	86 (±4.8)	59 (±4.3)	80 (±3.5)	97 (±1.5)	72 (±3.9)	78 (±2.6)

3.9 まとめ

本研究では 3 種類の枝刈り手法を提案し，ターン制ストラテジーにおける minimax 探索ベースの探索法に適用した．これらの手法は「重要な手の見落とし」のリスクを伴いながらも深く高速な木探索を実現させる．

そして，これらの枝刈り手法それぞれが人工プレイヤの性能に与える影響を調べるための予備実験を TUBSTAP プラットフォーム上で行った．その実験により，確かに性能を劣化させるリスクはあるものの計算時間を大きく短縮することを確かめた．最終的に本研究ではこれらの枝刈り手法全てを適用した人工プレイヤと，AI 競技会の優勝プレイヤとの対戦を行った．この実験によって提案プレイヤが優勝プレイヤを強さで上回ることを確認した．

提案手法はある程度の見落としのリスクは持ちつつも，UCT 探索ベースの既存手法たちと比べれば「浅い深さにある致命的な影響力を持つ読み筋」を見落とすリスクが少ないと考えている．そのため本手法は既存の研究に対する貢献として，着手候補の多いターン制ストラテジーの戦闘で重大な読み筋を見落としにくいアプローチを提案した点が挙げられる．

本章の枝刈り手法によって既存のターン制ストラテジーゲームの一部，そしてこれと似たような性質の「駒操作の手順前後可能性」や「類似した結果を導きそうな移動・攻撃行動のグループ作成可能性」を備えたゲームジャンルにおける強い人工プレイヤ実現の可能性が示された．それによって「ターン制ゲーム一般での強い人工プレイヤ作成」の実現に貢献したと考える．

第4章　複数戦略モンテカルロ法による RPGゲームのプレイヤ効用の推定

本章は会議への投稿論文を元に書き直したものである [36]．またその際の訳語は，先行研究となる論文 [37] [38] で用いられた表現を多く流用している．本章では，多様な対象と目的のゲーム人工プレイヤのための，ターン制ゲームにおける「強さ以外の目的」の人工プレイヤ作成に着目した研究について述べる．この「ターン制ゲーム・強さ以外の目的」の領域の中で，幅広い未解決な課題の中から本研究が対象に選ぶのは，RPG ゲームでの仲間役の人工プレイヤである．

4.1　RPGゲームと仲間役の人工プレイヤ

ゲーム情報学の基本的な目標の一つとして「人間を楽しませる人工プレイヤ」を作ることがあげられる．その目標を達成するために，まず人工プレイヤ研究の多くは強い人工プレイヤの作成を目的とし，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，人工プレイヤが人間のプロレベルの強さにまで達するための手法が多く開発された．そして複数のプレイヤによるチームとしての強さに焦点を当てた研究もされ，Sander らは QuakeIII のゲームプレイヤ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を提案した [70]．こうした強いゲームプレイヤは，人間プレイヤの「手強い敵」としての役割は十分果たせるかもしれない．

しかし，人間プレイヤの「良き仲間プレイヤの作成」に特化した研究は少ない．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，人間プレイヤの敵だけでなく仲間の立場を人工プレイヤがこなす場合がある．しばしば仲間役の人工プレイヤは人間の期待に外れた行動を取り，人間に不満を生じさせる．このような「仲間の期待外れな行動に対する」不満は，敵に期待外れの行動をされる場合の不満より大きくなりがちである．

このような期待外れが起こる原因として，この種のゲームに存在する “勝利以外の副目的” が理由として考えられる．単に勝率を最大化する行動は人間の副目的と抵触する恐れがあり，人工プレイヤは人間プレイヤの “どのような副目的をどの程度重視しているか” といった価値観を行動選択履歴より理解して，その価値観に沿った行動選択を行うべきであると本研究では考える．

本研究の目的は人間プレイヤの価値観を推定して不満を減らす仲間役の人工プレイヤ開発である．そのために人間の副目的を関数でモデル化し，複数戦略モンテカルロ法によって適切なパラメータの決定を行う．

4.2 関連研究

主に強い人工プレイヤ作成を 2 人対戦ボードゲームで目指す形の従来研究の場合と異なり，このゲームジャンルでは以下のような状況の違いが挙げられる．

- 一緒に遊ぶ人間プレイヤにとっての満足度が追求の対象となる点
- チームプレイが必要である点
- 人間プレイヤのモデリングが必要である点

このジャンルでは人工プレイヤの行動選択の人間らしさは重要な要素である．一般に人工プレイヤの人間らしさに関しては多くの研究や競技会が行われてきた [72]．例えば藤井らは生物学的制約を導入した学習手法を提案し，アクションゲーム Infinite Mario Bros. にてその性能を試験した [71]．また動作の自然さと似た概念だが，より複雑な概念として “believability” というものがあり，Matteo らはゲームキャラクタの「個性または動作の狙いの一貫性」が believability に非常に重要な影響を持つことを指摘した [74]．

ある種の多人数プレイヤによるゲームは，例えばサッカーのようなチームプレイが必要になる．こうした種類のゲームをを使った強い人工プレイヤ開発の研究は様々に試みられてきた．Bakkes らは敵チームに適用する進化的計算を導入してゲーム Quake III 上で人工プレイヤの実装と評価を行った [70]．

本研究が対象とするゲームも多人数のプレイヤによるもので，コマンドベースのロールプレイングビデオゲーム（RPG と呼ぶ）である．この種のジャンルでは例えば Wizardry や Final Fantasy シリーズが有名である．この RPG ゲームの多くでは，プレイヤのチームが敵（モンスター）のチームより強く設定されていることが多いが，しかし沢山の敵チームと連続して戦わされて，1 回ごとの戦闘終了時のキャラクタのステータスが次の戦いに一部引き継がれることがほとんどである．そのためプレイヤは戦いに「単に勝つ」だけでなく，より望ましい勝ち方を目指す．例えばキャラクタの体力や精神力といったステータス値をできる限り高く保ったり，時間の消費を抑えようと試みる．そうした，望ましい勝ち方のために問題となる要素を本研究では副目的と呼ぶ．そうした副目的のうちどれに重きを置くかは人間プレイヤの個々人によって様々で，例えば時間がかかっても安全に勝ちたいプレイヤもいれば多少危なくても早く勝負を終わらせたいプレイヤもいる．そしてこの例のように，価値観の違いによって望ましい戦闘スタイルが正反対になることもあるのでプレイヤの価値観を正しく推定することは非常に重要である．

Bakkes らのアプローチが “敵” の傾向を読み取って利用するのに対して本研究では “味方” の価値観を読み取り，その価値観に迎合した行動を人工プレイヤにとらせることを狙

う．そのために，味方人間プレイヤが過去にとった行動を観察する．プレイヤの行動を観察することでその個人の「行動選択への価値観」を模倣する試み自体はボードゲームでもよく見られ，特にプロのプレイヤの棋譜を利用し強い人工プレイヤ作成を行う際に，そうした試みがされる．$\alpha\beta$ 探索やモンテカルロ木探索の性能が，そうしたモデル化による行動評価関数の利用で向上することはよく知られている [75] [83] [85]．また，現在の相手プレイヤのモデル化と価値観の推定による「つけ込み」もよく試みられている [77] [78] [79]．

行動への価値観ではなく，状態（ゲーム局面）に対する価値観の模倣もよく行われる．保木らが人間の棋譜から状態評価関数を学習した手法もこの一種とみなせる [80]．また生井らが行った，ある特定個人の棋譜からそのプレイスタイルを再現する試みを行った研究も，ゲーム状態への価値観を模倣しているといえる [84]．本研究も同様にプレイヤ個人の価値観を読み取るが，２つの困難が想定される．つまり，多様な副目的の存在と，状態遷移のランダム性である．似たような試みには Y. Ng らによる inverse reinfrocement learning があり，彼らはエージェントの方策から報酬の関数を推測している [81]．

人間の複雑な価値観をモデル化するとき「効用」という概念 [82] がよく用いられる．例えば，ある人は「70% の確率で 1 万円がもらえるが 30% の確率で 1 万円を失うギャンブル」の話をもちかけられれば，合理的な思考の結果としてそれを受けたがるかもしれない．しかし似たような話の内容であっても「70% の確率で 100 万円がもらえてそれ以外の場合 100 万円を失うギャンブル」を彼が嫌がるであろうことは感覚的に十分想像できる．このような現象は，単なる期待値ではなく「効用」の理論を考えることでよく説明ができる．1 万円を得ることと失うことは多くの人にとって似たような重大さ（正負の方向性の違いはあれ）を持つが，100 万円の場合は事情が異なってくる．100 万円を得ることの嬉しさに対して 100 万円を失うことはより著しい負の方向の重大さを感じる人が多いと考えられる．こうした金額の値に対する精神的重大さの非線形的変化を効用関数で記述できる．ある 1 つの選択が複数の要素からなる場合，例えば車を 1 台買うときに付随する要素{値段，カーブの曲がりやすさ，速さ，頑丈さ，燃費の良さ} なども，効用関数によって記述できる．本研究の提案手法は人間プレイヤの効用関数を推測して利用することで良いチームメイトプレイヤとして振る舞うことを目指す．

4.3 接近法

人間プレイヤと協力して遊び，楽しんでもらえる RPG ゲーム人工プレイヤの作成を目指して以下の方法でアプローチした．RPG では人工プレイヤはしばしば人間の期待に外れた行動を取り，その期待と行動の不一致が蓄積して大きな不満につながりうる．

そうした不一致には様々な原因が想定できて，例えば勝利のための最善行動が見えづらいような複雑な状況であったり，人工プレイヤが勝利のための最善行動を選んでいても人間プレイヤが状況を誤解している場合などが考えられる．しかし不一致の原因として本研究が考える主要な理由は，副目的の存在である．例えばコンピュータプレイヤが勝利のために最善の行動をとっていても，ある人間プレイヤから見てあまりに勝つまでの時間が長

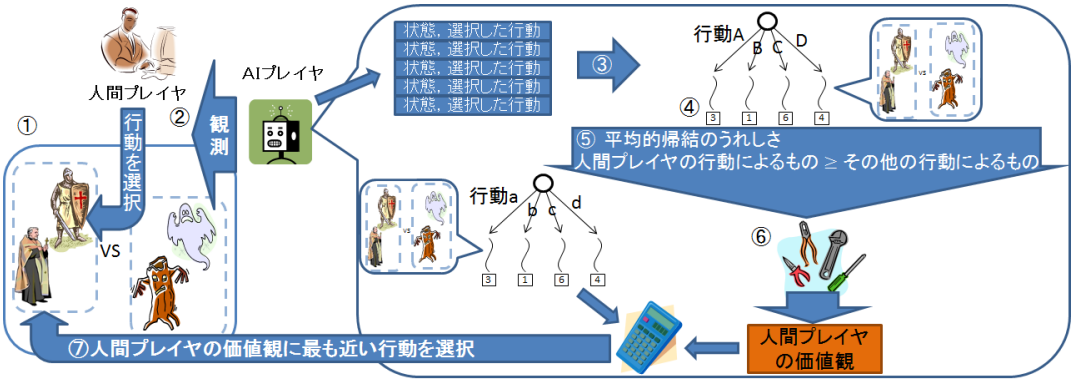

図 4.1: アプローチの全体像

かったり精神力を浪費しすぎる行動に感じられ，その行動を好まないかもしれない．その人間にとっては，多少勝ちの確率が下がったとしても，もっと敵を早く倒そうとする行動や MP を節約する行動を仲間にとってほしいと感じている可能性がある．よって，単なる勝ち以外の副目的を正しく読み取ってそれに迎合することが必要になる．

図 4.1 に提案手法の概要を示す．この節では図中の番号にあわせてそれぞれの手続きを簡潔に示す．

1. **対象ゲームの設定**：本研究では，複数のキャラクタからなる味方チーム及び敵チームが対戦するゲームを想定する．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し，主にコマンド形式 RPG を主眼に据える．主たる目的は各戦闘の勝利であるが，プレイヤはそれだけでなく “望ましい形の勝利” を目指す．つまりプレイヤには副目的があることを想定する．人間プレイヤにとっての “望ましい形の勝利” の厳格な定義は不明であり，また個人ごとに異なることを想定する．

2. **プレイヤ行動の記録**：ゲーム中の 1 キャラクタを人間プレイヤが操作するものと想定する．キャラクタの各選択行動は，それを選んだときのゲーム状態と対にした形で記録され，価値観推定のために利用される．

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．その推定に基づき人工プレイヤは人間にとって都合の良い行動を選択し，また以後も戦いの進行に伴って行動のデータが更に蓄積していくので推定は随時更新されていく．

4. **平均的帰結のシミュレーション**：各状態で人間は候補となる行動の中から 1 つだけを選択する．その際に各行動の候補がどのような結果を将来招くのかについて，人間はある程度予測した上で選択を行っていると考えられる．そのような「各行動の選択後に訪れる結果」をモンテカルロシミュレーションにより推測する．このような推測を，実際に人間に選ばれた行動だけでなくその他の行動の候補についても行

う．そして多数のシミュレーションによる結果を平均したものを各行動の**平均的帰結**とする．

5. **帰結の解釈**：人間プレイヤが選ぶ行動は，その人の価値観から見て最も好ましい結果にチームを導く，と考えるのが妥当である．言い換えれば「人間プレイヤの選んだ行動の平均的帰結は，その他の行動候補の平均的帰結と比べて人間プレイヤにとって嬉しいものである」と解釈できる．

6. **価値関数の推定**：各プレイヤは「どのような対結果をより好ましいと感じるか」を決める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**：推定によって効用関数のパラメータを定めた後，人工プレイヤは意思決定の際に可能な行動がそれぞれどのような平均的帰結をもたらすかをシミュレーションする．そしてそれぞれの帰結が効用関数によりどの程度好まれるのかを計算して，その中で最も好ましい帰結を導く行動を選択することで人間プレイヤに迎合する．

このようにして人間プレイヤの行動の観察から，人工プレイヤがとるべき望ましい行動が提案される．もちろん人間プレイヤが自分で目指す行動の指針と仲間にとってほしい行動の指針が食い違う場合も考えられる（例えばゲームの特殊なルールによっては，敵を倒したキャラクタのみに与えられる報酬欲しさに，人間は「自分は積極的に攻撃する」かつ「仲間プレイヤは攻撃を控える」状況を好ましく感じる場面が想定できる）．しかし本研究の手法は結果局面をより精密に解釈する（例えば「誰が敵を倒したのか」も帰結の効用計算に利用する情報に含める）ことで，そうした状況でも適切な行動を選ぶ人工プレイヤが作れると考える．

4.4 アルゴリズム

この節では本研究のアプローチの流れに沿ったアルゴリズムの全体像を見せる．適宜使用される記号の意味を表 4.1 にまとめる．

4.4.1 プレイヤ行動の記録

今回の研究ではマルコフ決定過程 [76] によって対象ゲームをモデル化する．S を状態の離散集合，A を行動の離散集合とする．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の履歴を蓄積して推定に用いる．人間プレイヤの j 回目の行動選

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
$\pi : S \to \mathbb{R}^{\|A\|} \in \Pi$	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

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 $a^j \in A_{s^j}$ とここで表記する．そのペアの集合 $\{(s^j, a^j)\}^j$ を履歴として記録する．

4.4.2 平均的帰結のシミュレーション

平均的帰結の導出にモンテカルロシミュレーションを用いる．各シミュレーションは，着目する行動選択直後の状態から始まって戦闘の終結まで行う．シミュレーション中の行動選択は，「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 $s_i(s, a, \pi)$ は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 n 次元の実数値ベクトル $\vec{x}_i(s, a, \pi)$ を得ることにする．その後 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として以下を求める．

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

各行動を等しい確率でとるような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戦略で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 [85]．そのような偏ったシミュレーション戦略では各プレイヤからみて「良い行動」が積極的に選ばれ，チェスや囲碁のように「行動の良さ」が比較的単純に定義できるゲームでは偏った戦略を 1 つだけ採用すればよい．

それに対して RPG では副目的が様々あって，何が「良い行動」であるかは各プレイヤの価値観によって変化する．そのため本研究では偏った戦略を複数用意する．そのうえで図 4.2 のように，各戦略 π ごとに平均的帰結 $\vec{x}(s, a, \pi)$ を別々に計算する．

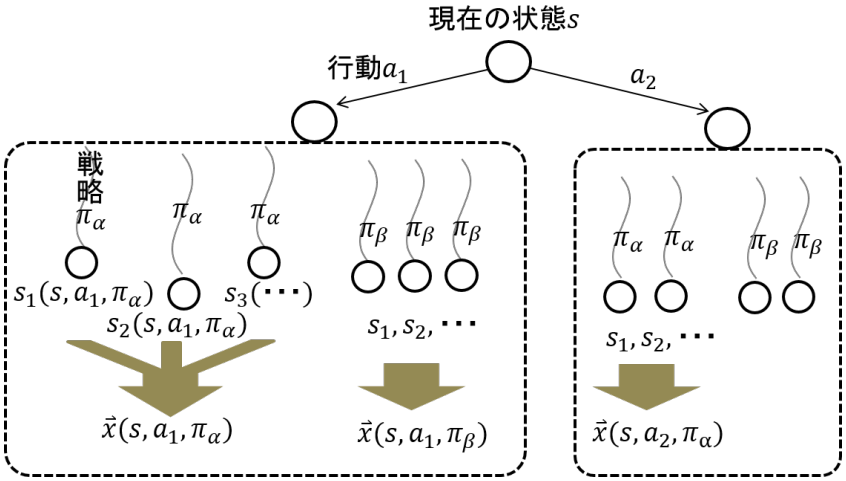

図 4.2: 平均的帰結のシミュレーション

本研究ではこのような複数戦略によるシミュレーションを，平均的帰結を求めるためだけでなく人工プレイヤの行動決定時にも用いる．

4.4.3 価値関数の推定

本研究では効用理論に基づいて，人間プレイヤはそれぞれ独自の効用関数を最大化するように行動選択を行っていると仮定する．その効用関数の具体的な形は不明なので，パラメータ化された何かしらの関数モデルを用意しパラメータの最適化を行う必要がある．

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．そのモデルによる効用関数 $u : \vec{x} \to \mathbb{R}$ 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 による平均的帰結の特徴量ベクトルであり $\vec{w}$ は重みベクトルである．

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」ということは，「a^* による平均的帰結は，その人間にとって他の行動による平均的帰結より望ましいものである」という解釈ができる．よって，ある戦略 π を可能な戦略の集合 Π の一要素として以下の不等式 (4.3) が成り立つことが期待される．

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

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られる．よって本研究ではこの式を満たさない行動履歴データの数が最小になるようなパラメータを探す．

そこで本研究では，人間プレイヤのありうる重みベクトル空間 $W \ni \vec{w}$ 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)

Algorithm 1 プレイヤの重みベクトル推定アルゴリズム

```
for each w⃗ ∈ W do
  p_w⃗ = 0
end for
for each (s, a*) ∈ {(s^j, a^j)}^j do
  for each w⃗ ∈ W do
    u* = max_{π∈Π} u(x⃗(s, a*, π), w⃗)
    for each a ∈ A_s \ a* do
      if u* < max_{π∈Π} u(x⃗(s, a, π), w⃗) then
        p_w⃗ += 1
      end if
    end for
  end for
end for
return arg min_{w⃗∈W} p_w⃗
```

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，候補となる全ての重みベクトル $\vec{w} \in W$ 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．また，もしも候補となるベクトルが多数ある場合は，それらの平均となるベクトルが選択される．

こうした推定のためのアプローチとして，勾配降下法もまた効果が見込めそうな手法の 1 つである．本研究の現在の，空間の離散化による手法は扱う次元の増大に従って計算コストの爆発的な増大が予想されるが，勾配降下法はそうした状況でもうまく働くことが予想されるためである．

4.4.4 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．本研究ではそのような行動選択を行う人工プレイヤを “モンテカルロプレイヤ” または “MC プレイヤ” とここでは呼ぶ．

4.5　本研究で扱うゲームの概要

本研究では提案手法の実装や実験のために，学術的な再現性を考えながら独自にコマンド形式のターン制ゲームを設計し，その上で手法の性能評価も行った．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．

4.5.1　キャラクタの持つパラメータ

このゲームは 2 チーム対戦の形式で，各チームは複数のキャラクタからなる．ゲームに参加するキャラクタには以下の 4 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．市販の RPG ゲームでは敵のパラメータの一部または全てが未知であることが多いが，そうしたゲームでも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．よって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，全キャラクタのパラメータは完全情報として扱うことにした．

- 体力（HP）: 体力は敵からの攻撃により減少する．体力が 0 になるとそのキャラクタは行動不能になって，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．またゲーム開始時の体力を「最大体力」といい，後述する回復などの特技はキャラクタの最大体力を超えない範囲で HP の回復を行う．
- 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．高いほど大きなダメージを与える．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．高いほど受けるダメージが減少する．固定値．

4.5.2　行動とその効果

各キャラクタの行動として，使用する術技と対象キャラクタを選ぶ．例えば “単体攻撃を敵 1 に行う”，“中回復を仲間に行う” などが各キャラクタの行動である．各術技の効果を以下に述べる．

- 単体攻撃 : 対象の敵キャラクタに（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．

図 4.3: ゲームの流れ

- グループ攻撃 : 敵チームはときどきキャラクタ数体からなるグループを形成するが，敵 1 グループを対象として，グループ内のキャラクタそれぞれに 30 のダメージを与える．精神力を 8 消費する．
- 小回復 : 対象キャラクタ 1 体（または自分自身）を対象にして，体力を 42 回復させる（最大体力は超えない）．精神力を 4 消費する．
- 中回復 : 体力の回復量と消費する精神力の量が小回復より大きい．体力を 88 回復させ，精神力を 8 消費する．
- 全体回復 : チーム全員の体力を 160 回復する．精神力を 18 消費する．
- 防御 : 次に自分の番がくるまで受けるダメージ量を半分にする．

4.5.3 状態遷移

ゲームの大まかな流れを図 4.3 に示す．戦闘不能になっていない全てのキャラクタは毎ターン一度ずつ行動できる．行動の順番はランダムに決まり，プレイヤはその順序を知ることはできない．

市販の RPG ゲーム，例えばドラゴンクエストや Wizardry シリーズでは，ターンの開始時に全ての味方チームの行動を決定しなければいけない．しかし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．よって，ひとまずは本研究ではより単純な状態遷移ルールのゲームを扱う．

本研究の設計したルールではキャラクタの行動は，ターンの開始時ではなく行動の順番がまわって来た時に行動選択を行う．そして行動を選択するとすぐにその行動結果を反映する形で状態が遷移し，次のキャラクタに行動選択権が移る．戦闘不能になったキャラクタには行動の順番は回ってこず，味方か敵のチームのキャラクタが全滅するまで行動選択を繰り返す．

表 4.2: 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

4.6　戦闘参加キャラクタの設定

チェス等の古典的な 2 人対戦ボードゲーム等と異なり，RPG ゲームでは普通，互いのプレイヤチームの戦力に大きな差がある．更に RPG では大きくパラメータの異なる様々なタイプの敵チームと戦う．それを踏まえて本研究では手法の評価のために，5 つの異なる設定の戦闘状況を用意した．弱い敵を倒すだけの状況から，強力な “ボスキャラクタ” との戦闘を想定した設定もある．

それらのうちの設定 2 を表 4.2 に示す．この設定では，小回復ができる敵を含む敵チームを相手に戦う．勝利自体は簡単であるが，MP を温存して勝つことは難しい．というのも，単純に MP を一切使わずに戦闘を進めようとすると敵の小回復のために勝負が長引き，こちらも回復を使用せざるを得ない状況になって MP の消費がかさんでしまう．そのため MP を温存するためには，序盤から MP 消費を伴う強力な術技をふんだんに使って敵の回復役をなるべく早く倒す必要がある．このような一見矛盾した戦略は人工プレイヤにとって理解しづらく，人間の効用を推定する作業をある程度難しくすることが予想される．

4.7　特徴量と重みベクトル

提案手法では，戦闘のシミュレーション結果（平均的帰結）から n 次元の特徴量ベクトルが抽出されて，n 次元の重みベクトルによってその結果が評価される．本項では，特徴量ベクトルの作り方と，また効用関数の重みベクトル $\vec{w}$ が変化するとどのように人工プレイヤの行動選択が変わってくるのかを説明する．

4.7.1　特徴量ベクトル

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，そこから情報を 3 種類のみ抜き出して 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とメモリのサイズも増える．

本研究における平均的帰結 $\vec{x}$ の特徴量ベクトルを式 4.4 に示す．

$$\vec{x} = \{x_{HP}, x_{MP}, x_{Turn}\} \tag{4.4}$$

式 4.4 に示された各特徴量の計算式を式 4.5〜4.7 に示す．

$$x_{HP} = \frac{\text{自チーム全体の残り HP}}{\text{自チーム全体の最大 HP}} \tag{4.5}$$

$$x_{MP} = \frac{\text{自チーム全体の残り MP}}{\text{自チーム全体の最大 MP}} \tag{4.6}$$

$$x_{Turn} = b - \text{経過ターン数} \times a \tag{4.7}$$

ここで，a, b, は設定ごとに異なる定数である．もし味方チームが少ないダメージで勝利した場合，x_{HP} の値は大きくなり，精神力の消費を抑えて勝利した場合は x_{MP} が大きくなる．そして味方チームが少ないターン数内で勝利をおさめた場合は x_{Turn} が大きな値となる．ここで，ゲームの主目的である「勝利」の情報が x_{HP} の値に織り込まれていることに注意されたい．というのも，もしも x_{HP} が最低値である 0 となっている場合その戦闘は敗北であり，正の値なら勝利している．

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 の温存を重視する効用，[1,0.1,0.1] は HP の残量を重視，つまりダメージをなるべく避ける効用，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．つまり W は可能な重みベクトル 961 個を含む．

4.7.3 効用重みが行動に与える影響

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．重みベクトルの値の変化にともなって人工プレイヤの選択する行動も適切に変わることが望ましい．例えば「MP 重視の重みベクトル」を持つ人工プレイヤが MP を温存した勝利を目指す行動をきちんと選ぶことが望ましい．

さて，ここで 2 つの重みによるモンテカルロプレイヤ（MC プレイヤ 1 と MC プレイヤ 2 とする）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．この値は以下の手続きで計測される．

1. MC プレイヤ 1 が 1 体のキャラクタを操作し，k 個の状態行動ペア $\{(s_i, a_i)\}$ を記録する．

2. 状態 s_i それぞれ，MC プレイヤ 2 が同様のキャラクタを操作して，選択された行動 a'_i を記録する．

3. その a_i と a'_i が一致した回数を k で割ったものを “行動一致率” とする．

ここで，全く同じ重みベクトルを MC プレイヤ 1 と 2 が持っていたとしても，モンテカルロ法の乱数性により行動全てが一致するとは限らないことに注意されたい．

設定 2（表 4.2）の味方 2 を人工プレイヤが操作する状況を与える．ここで，様々な重みベクトルの MC プレイヤと $\vec{w} = [1, 4, 8]$ の重みベクトル（MP と Turn 重視）の MC プレイヤの行動一致率の分布を図 4.4 に，様々な重みベクトルのプレイヤと $\vec{w} = [1, \frac{1}{8}, \frac{1}{16}]$（HP 重視）の重みベクトルの MC プレイヤの行動一致率の分布を図 4.5 に示す．この 2 つの図では傾斜を見やすくするため互いに軸は逆向きに表示している．

$\vec{w} = [1, \frac{1}{8}, \frac{1}{16}]$ の重みベクトルを用いた MC プレイヤを用いた場合では，同様の $\vec{w} = [1, \frac{1}{8}, \frac{1}{16}]$ なる重みベクトルを用いたとき行動一致率が最も高くなっている．しかしその周辺にも一致率が 70% を超えている丘状の領域がグラフに見られる．一方で $\vec{w} = [1, 4, 8]$ の重みベクトルを用いた場合では切り立った崖上の分布が見られ，重みが $[1, 4, 8]$，$[1, 8, 16]$，$[1, 16, 32]$ だと一致率は高いがそこから少し値が外れただけで一致率は大きく低下する．この結果より，重みベクトルの値が多少ズレていてもほとんど似たような行動選択が行われる場合があることが解る．そのため本研究では，人工プレイヤの効用推定の精度について，学習された重みベクトルそのものよりも行動一致率をより適切な評価の尺度として用いる．

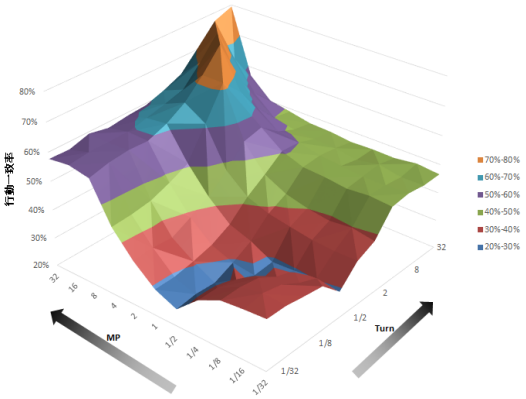

図 4.4: $[1, 4, 8]$ の場合の行動一致率の分布

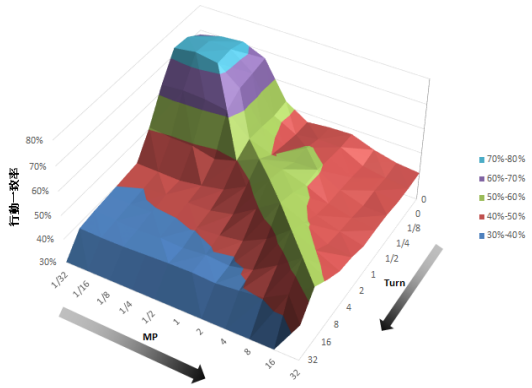

図 4.5: $[1, \frac{1}{8}, \frac{1}{16}]$ の場合の行動一致率の分布．前図とは各軸の向きが逆．

次に本研究では，例えば MP を重視するような重みベクトルが本当に「MP 消費を抑えた形での勝利」を導くのかを確かめるための，簡単な実験を行った． 3 種の典型的な

表 4.3: 3 種の典型的重みベクトルによる戦闘結果．最後の行については 4.8.2 節で説明する．

戦略	重み	HP 合計	MP 合計	ターン数
複数	HP 重視 (**1**,0.1,0.1)	**238**	41	8.8
複数	MP 重視 (1,**10**,1)	150	**80**	11.5
複数	ターン重視 (1,1,**10**)	160	56	**6.2**
単一	MP 重視 (1,**10**,1)	159	63	6.9

重みベクトルを用意し，そのどれかを持ったモンテカロルプレイヤが味方 2 を操作した．1000 回の戦いの結果，得られた平均的帰結を表 4.3 にまとめた．その結果により，HP 重視，MP 重視，ターン数重視の重みベクトルを用いるとそれぞれ，HP，MP が最も高い帰結，勝利までのターン数が最も少ない平均的帰結が得られたことを確認できた．

4.8 複数戦略モンテカルロ

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研究の特色の一つである．単一のランダムなシミュレーションが行われるモンテカルロ法と比べて，複数戦略モンテカルロでは個々の戦略ごとに割り当てられるシミュレーション数は減少するが，引き換えにシミュレーションがより現実的になる．この節では，本研究が用意した 7 つの戦略と，その効果を示すための実験について述べる．

4.8.1 戦略の設計

対象ゲームに応じた複数の戦略を用意する．どの「行動とそれ以降の戦略」が採用されれば最も望ましい帰結が得られるのかがモンテカルロ法により選ばれる．そして個々の戦略全てを注意深く設計する必要はない．というのは，アルゴリズムは最も適切な行動と戦略の組み合わせを選ぶので，あまりに愚かな戦略が含まれていたとしても単に選択されないだけで害はない．とはいえもちろん計算資源の消費にはつながるのでそういう愚かな戦略はもし可能なら最初から除外されるべきである．

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：味方の残り HP が高いほど HP 回復系の術技の選択確率を低くする．通常，HP が高い状況での HP 回復は悪い着手である．
3. 攻撃重視：単体およびグループ攻撃系の術技の選択確率を他の行動の 5 倍にする．
4. 単体攻撃重視：単体攻撃の選択確率を他の 5 倍にする．MP の節約に向く．

表 4.4: 複数/単数戦略使用時の HP 重視/MP とターン重視の重みによる勝率の変化

重み	複数戦略	単一戦略
HP 重視 (1,0.1,0.1)	98.8%	96.0%
MP 重視 (1,10,10)	83.6%	70.2%

5. グループ攻撃重視：グループ攻撃術技の選択確率を 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．

4.8.2 予備実験：複数戦略の効果の検証

ボードゲームでは，シミュレーション戦略の良さが人工プレイヤの強さに貢献することが知られている [85]．この節で本研究では，RPG ゲームにおける複数戦略の採用が単なる強さの向上だけでなく，特徴的な終局状態への誘導にも役立つことを実験で示す．

まず初めに表 4.5 に示す設定 5 を用いて，単一戦略と複数戦略のモンテカルロプレイヤの勝率比較を行った．味方 2 を単一戦略または複数戦略のモンテカルロプレイヤが操作し，味方 2 以外のキャラクタはルールベースで動作する．効用の重みベクトルを 2 種を用いて，1000 回のゲームプレイで味方 1 と 2 が共に生き延びたゲームの割合を計測した．その結果を表 4.4 に示す．この勝率（味方が生き延びた割合）だけに着目するのなら，HP 重視戦略の方が他の戦略よりも良い結果を残し，さらに単一戦略よりも複数戦略の方が更に良い結果を達成している．

次に，単一戦略と複数戦略で，効用の内容を適切に反映するような行動を選ぶ能力の比較を行った．重みベクトル（1,10,1) の MP 重視戦略の単一戦略モンテカロルプレイヤのテスト結果は前項と同様の条件でゲームを行い，その結果も表 4.3 の最も下の行に示されている．単一戦略は複数戦略の場合に比べて，MP 残量が低くなってしまっている一方で消費ターン数が少なくなっている．つまり複数戦略の方が，効用を反映した状態に局面を誘導しやすい．

個々のゲームを検証してみると，単一戦略モンテカルロプレイヤはしばしばグループ攻撃を序盤に多用してしまっていた．その原因は，単一戦略では（ランダムな）シミュレーション中が行われて，MP 消費を伴う行動がかなり頻繁に使用されるためである．このような場合では，最終的な帰結における MP 残量が「序盤の MP 消費量」よりも「乱数の偶発性」に大きく支配されてしまう．一方で複数戦略モンテカルロの場合は，用意された戦略（例えば「単体＋適時回復」）によって，シミュレーション中でも MP を温存する行動を重点的にとることができる．よって最初に選んだ行動が帰結にもたらすことができる影響を正しく検知できる．

4.9　人工プレイヤに対する学習実験

本稿では，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を実験で確かめる．効用を推測される側の人工プレイヤに持たせた効用関数重みベクトルは $(1, 0.071, 0.071)$，$(1, 0.143, 18)$，$(1, 12, 0.167)$，$(1, 10, 10)$ の 4 通りである．また敵味方のキャラクタ設定は表 4.2 と 4.5 に示す計 5 通りで，全 20 パターンの場合を調べた．実際の RPG で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．

全ての場合において味方 2 のみが「効用関数を持たせた人工プレイヤ」に操作され，味方 1 や敵はルールベースの単純な動作を行う．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（ただし HP と MP もその都度初期化される），1 回目の半分，1 回目終了時点，3 回目，5 回目，8 回目の戦闘終了時点までで蓄積したデータで効用関数の推定を行う．また提案手法のモンテカルロシミュレーションは乱数に大きく影響を受けるため，この 8 回の戦闘を 1 セットとし，乱数のシード値を変えながら 20 セット（計 160 戦闘）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，行動一致率（得られた効用重みによって取る行動が元の効用関数によるものと一致する割合）を用いる．これは，例えば図 4.5 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 4.6 には，持たせた効用関数の重みごとの行動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，学習の限界点ともいえる．どの設定の場合でも，8 戦後の効用重みによる行動一致率と限界点の差はたかだか 3% 程度であり，提案手法は戦闘の設定に関わらずに安定して高い結果を発揮したと結論できる．

4.10　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が用いられた．そこで次は，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．

4.10.1　実験条件

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．戦闘の状況は表 4.2 の「設定 2」のみを使う．次に戦闘 8 回を 1 セットにして，4 セットの戦闘を行ってもらった．各セットでは被験者に以下の指示を与えた．

- なるべく HP を高く保ちながら戦う

表 4.5: キャラクタのパラメータ　設定 1，3，4，5

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
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
設定 4					
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
設定 5					
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

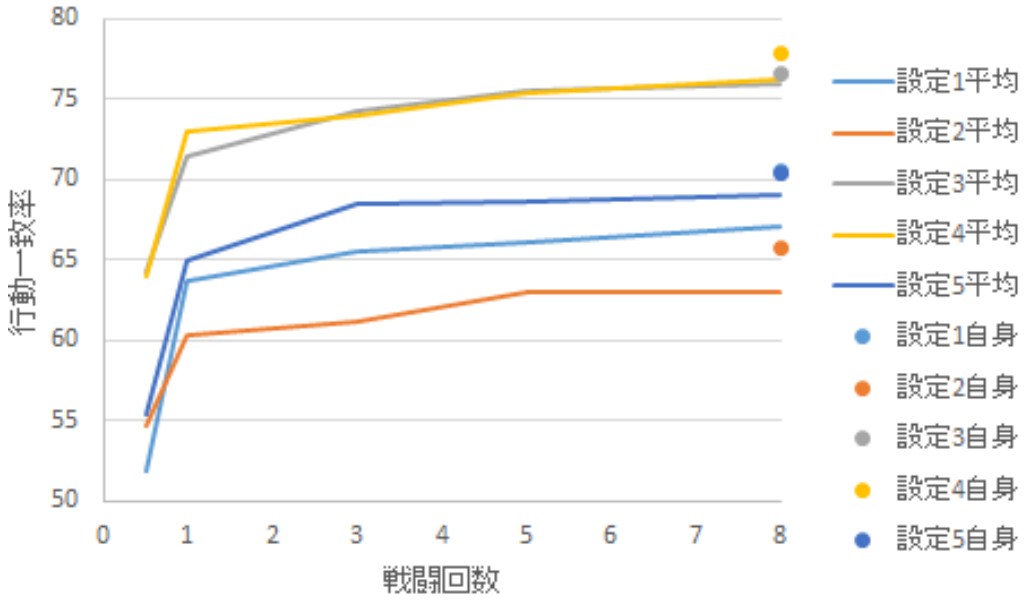

図 4.6: 行動一致率・戦闘の設定別

- なるべく MP を温存しながら戦う
- なるべく早いターン数で勝ちを目指す
- なるべく早いターン数で，なおかつ MP も温存しながら戦う

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 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 1 のみ操作してもらい，AI が操作する味方 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 種類で，一つは提案手法（効用関数を推定した AI），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．計 4 回の戦闘のうち，提案手法 AI は 2 回，他の AI は 1 回ずつ戦う．AI の登場順はランダムである．また今回の実験では x_{HP} の計算法を少し変えた．ゲーム終了時の HP ではなく戦い全体を通じての平均 HP 量を x_{HP} の計算に用いた．こうすることによって，人工プレイヤが「ゲームの終了時のみ体力を慌てて回復する」ようなことを避け，ゲーム全体を通じて体力を高く保って安全に戦うという価値観に近い挙動が得られると考える．

4.10.2 結果

いずれも RPG ゲームの経験を持つ被験者 10 人による自然さの評価値の平均を表 4.6 に示す．実験は Windows の GUI プログラムを通じて行われ，それぞれ 1 時間から 2 時間程度を各被験者が要した．

表 4.6: 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
HP 温存	提案手法	3.8
	MP 重視 AI	3.2
	Turn 重視 AI	2.9
MP 温存	提案手法	3.4
	MP 重視 AI	3.0
	Turn 重視 AI	2.1
速い勝利	提案手法	4.2
	MP 重視 AI	2.5
	Turn 重視 AI	4.0
速い勝利かつ MP 温存	提案手法	4.0
	MP 重視 AI	3.0
	Turn 重視 AI	2.7

表の数値をみると提案手法がどの場合も固定重みの人工プレイヤより良いスコアを獲得している．例えば MP 重視を指示した場合，MP 重視 AI に対する評価 (3.0) は Turn 重視 AI に対する評価 (2.1) よりも高いが，提案手法はそれ以上の評価 (3.4) となった．学習された重みは [1, 20, 0.025] などより極端なものであり，固定で与えた [1, 4, 0.25] 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．

最後に与えた指示は速さ重視かつ MP 重視という幾分漠然としたもので，人間被験者によるさじ加減もまちまちだった．そのため推定された効用重みベクトルも [1, 11, 24] から [1, 23, 23] までばらついて分布した．このように提案手法は，単一の種類（指示）ながらも個々人でバラつきがある効用にもそれぞれ対応できるため，そうした状況が頻繁に生じるような実際のゲームでも有用であると考える．

4.11 まとめ

RPG ゲームでは単なる勝利以外の副目的が人間プレイヤの内に暗黙に存在すると考えることができる．RPG で人間らしいふるまいを追求する既存研究はあったが，人間個人の価値観にもとづいて不満の少ない行動を選ぶような研究は新規である．本研究では，プレイヤの行動選択からその背後にある副目的を「効用」として推定して，うまく相手に迎合するための手法を提案した．その効用はパラメータつきの関数でモデル化されて，複数戦略モンテカルロ法により各個人ごとのパラメータ最適化が行われる．提案手法の有効性は，人工プレイヤと人間プレイヤを用いた 2 つの実験により確かめられ，人工プレイヤの

効用重みを正しく推測できることと人間プレイヤと不満なく協力して戦闘できることが示された.

第5章　格闘ゲーム

本章は会議への投稿論文を元に書き直したものである [39]．本章では，多様な対象と目的のゲーム人工プレイヤのための，リアルタイム制ゲームにおける強い人工プレイヤ作成に着目した研究について述べる．この「リアルタイム制ゲーム・強さ目的」の領域の中で，幅広い未解決な課題の中から本研究が対象に選ぶのは，格闘ゲームである．

5.1　はじめに

強いゲーム人工プレイヤ開発のために様々な技術が開発されている．あるゲームではAI は人間の上級プレイヤーより強いが，あるものではそうではない．例えばチェスの分野では Deep blue が人間のチャンピオンプレイヤーに勝利している [1].

その一方で人工プレイヤが人間より強くないゲームの中の一つには，互いの行動が同時に処理される対戦型のゲームがあり，そうしたゲームはプレイヤーが交互に行動するタイプのゲームとは違う要因を考慮する必要が生じる．互いの行動が同時に起こるゲームでは多くの場合，最善行動が 1 つに限定できず，相手によって選択される行動に対応してプレイヤーの最善行動が決まる．もちろん相手がどの行動を選択するのかはそのプレイヤーにとって未知であり，ゲームも自ずと不完全情報ゲームとなる．

さてこのような同時行動型の不完全情報ゲームの一例として本研究では格闘ゲームに着目する．格闘ゲームとは 2 人のプレイヤーが各自のキャラクターを操作して互いに格闘させる形式のゲームであり，数多くの商業的なタイトルがビデオゲームとして世界中で販売されている．また格闘ゲーム人工プレイヤのための競技会 [5] が毎年開催されるなど，数多くの研究があるが既存のほとんどのゲームにみられる格闘ゲーム人工プレイヤは人間の上級者より強くない [86].

さて，その競技会の多くのプレイヤを大雑把に 2 つに分けると，ヒューリスティックな If-then ルールによるルールベース型のプレイヤと，そのときの対戦を通じて相手をモデリングして対応するオンライン学習型のプレイヤに分けられる．ヒューリスティックによるルールベース型プレイヤ（以下，単にルールベース型と呼ぶ）は格闘ゲームでかなり強い．なぜならば格闘ゲームは時間や相互の距離のスケールが細かいシステムであり，効果的な連続した行動を機械学習や木探索などで獲得するよりもヒューリスティックによりハンドコーディングする方がはるかに容易だからである．

実際に競技会で 2013 年と 2014 年の 3 C カテゴリーマッチで優勝したプレイヤはこのルールベース型である．その一方で，このタイプのプレイヤは，内部の If-then ルールを

相手の過去の行動履歴を含めるなど高度に洗練させない限りは，挙動が一貫して同じになる点が欠点である．たまたま相性の悪い相手に対して，ずっと同じパターンのおちいって単純に一方的に負けることがあり，実際の大会でも強いルールベース型が弱いルールベース型に大敗するケースが見られた．またこうした単調さは，格闘ゲーム人工プレイヤが人間と戦う場合にも問題になると考える，なぜなら同じパターンの動きが人間を退屈させうるためである．

それと対照的に，オンライン学習型のプレイヤは現在の相手に動的に対応して自らの挙動を相手に有利になるような意図の上で変化させる．しかし既存のオンライン学習型プレイヤは効果的な連続した行動をオンラインに獲得していくことが概して苦手である．

そこで本研究では，複数の既存のルールベース型プレイヤをそれぞれ“コントローラー”（詳細は5.3.1項にて述べる）として用い，コントローラーを組み合わせて切り替える方法を考えた．これにより，ルールベース型プレイヤの持つ効果的な連続行動に基づき相手にオンラインで有利に対応しようとする人工プレイヤの実現が期待できる．一定時間ごとに複数から1つのルールベース型プレイヤにキャラクターをコントロールさせ，現在の相手に対して不利なプレイヤはより少ない機会キャラクターをコントロールし，有利なプレイヤはより多い機会コントロールする．結果として本研究の提案手法による人工プレイヤは現在の相手に有利なヒューリスティック行動を多くとる．

考慮するべき点の1つは，どのプレイヤにどれほど長くキャラクターのコントロール時間を割り当てるかという問題である．本研究ではこの割り当てを古典的なExploration - Exploitation問題の一種である非定常Multi-Armed Bandit (MAB) 問題 [91] の一種として捉え，有力な既存手法であるSliding window upper confidence bound（SW-UCB）アルゴリズム [90] を用いた．

5.2 背景

この節でのみ本研究ではゲーム人工プレイヤを“AI”と呼ぶことにする．多くの研究者が彼らの人工プレイヤを文献中で“AI”と呼ぶためであり，そのため本研究も説明の混乱を避けるためその呼称を使う．

5.2.1 格闘ゲーム

格闘ゲームは，主にビデオゲームとして遊ばれる2人対戦ゲームである．図5.1のように，各プレイヤーによって操作されるキャラクターが互いにダメージを与えながら格闘する．このゲームは一般にリアルタイムであり，同時着手ゲームの連続とみなせる．可能な行動（着手）は通常，じゃんけんのような3すくみの関係を含む．例えばあるゲームでキック攻撃は投げ攻撃に強く，投げ攻撃はガード動作に強く，ガード動作はキック攻撃に強い．

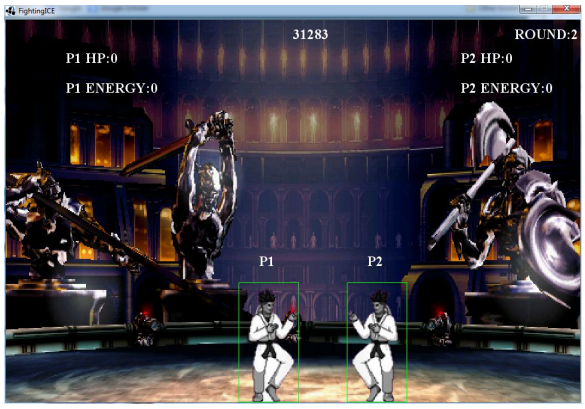

図 5.1: FightingICE スクリーンショット

このため，その 3 種の行動を適切な配分で実行し続けることが最適戦略にみえるかもしれない．しかし格闘ゲームの最適戦略の獲得は実際には簡単ではない，なぜなら通常格闘ゲームのシステムはじゃんけんより大いに複雑で，可能な行動は 3 より多くあり，キャラクターの相互距離や状態といった要素により行動間の関係性も変わる．そのため各プレイヤーの戦略には特定の偏りがある．それゆえ格闘ゲームでオンラインな機械学習により相手のモデル化に成功して相手の次行動を予測可能にすることは，プレイヤーを強くすることに貢献すると考えられる．

5.2.2 既存の格闘ゲーム AI 手法

格闘ゲーム AI のデザインには様々なものが考えられるが，本研究では以下の 3 つ型を主に考える．

- ルールベース型
- 機械学習型
  - オフライン学習
  - オンライン学習

このうちもっとも簡単なのは If-Then ルールの集合によるルールベース型である．これはゲームの状況があるルールの前提条件を満たせばそのルールによって定められた特定の動作を AI に行わせる．一方で機械学習はゲームの状態をインプット，ゲームへの入力キーをアウトプットとする各種モデル，例えば Q 学習にみられるようなテーブルや人工ニューラルネットワークなどを教師データや環境からの報酬に基づいて調整していく．

オフライン学習型では AI の挙動は大量のデータにより決定され，実際の使用時には試合を通じて挙動が一貫的である．この種類の手法による研究には Sarayut らによる，人

間プレイヤーの行動データの教師なし学習のクラスタリングを伴う Finite State Machine の構成による戦略模倣 AI の研究がある [88]．また Hyunsoo らによる，ゲームに関する訓練データからの事例ベースによって行動を決定する AI もこのタイプである [87]．

一方でオンライン学習型はオフライン学習型と違って，現在の相手や状況に対して有利な振る舞いを対戦中に獲得しようとする．オフライン学習により得られた初期値としての行動パターンを異なる手法でオンラインに拡張したものと，一貫して同じ手法で学習するものがある．前者は模倣学習の分野で見られ，人間によるプレイログを学習して AI の初期の挙動を決定した後，異なる方法でオンラインな性質を持たせる．Sarayut らは現在の相手に有利な行動の出現頻度を増やし [93]，星野らは対戦相手の戦略のうち自分の戦略と類似したものの模倣 [94] によって AI の性能向上を試みた．

オンライン学習で後者のように一貫して同じ手法で機械学習を続ける格闘 AI の研究は数多くある．強化学習では Thore らは AI 設計に SARSA 手法を用い [92]，Simon らはモンテカルロ法を用いた [89]．Eyong らは人工ニューラルネットワークの最適化による AI 設計を行った [95]．また Kaito らは k-NN 法による相手の行動予測 AI [86] もオンラインに学習を行う．

5.2.3 FightingICE

FightingICE [5] は 2013 年に公開された格闘ゲーム AI のプラットフォームであり，本研究の実験もこれを用いて行う．人間プレイヤー同士，AI プレイヤ同士そして AI プレイヤと人間プレイヤの対戦がそれの上では可能である．そして AI 開発のためのツールキットも公開されており AI 同士のコンペティションが毎年開催されている．

このプラットフォームは市販タイトルよりも簡略化された環境ではあるが，通常の市販される格闘ゲームが共通して備えている基本的な要素のほとんど，例えば攻撃，ガード，移動，リアルタイム性，コンボ，スペシャルアーツを持っていて，さらに AI には人間プレイヤーの反射神経を模倣した 0.25sec の認識遅れが課されていて，最適戦略を自明に構成するには十分複雑すぎる環境である．さらに，過去のコンペティションの出場 AI が公開されているためある提案 AI の性能評価も行いやすい．よって本研究ではこれを実験環境として選択した．

5.2.4 出場 AI にみられる設計

格闘ゲーム AI の設計法の中で，FightingICE の格闘ゲーム AI コンペティションの出場 AI の観察を通じて，頻繁にみられる以下の 2 つの AI 設計に着目する．ルールベース型と，機械学習のオンライン学習型（以下，単にオンライン学習型と呼ぶ）である．それぞれへの考察をこの節で与える．

オンライン学習型

オンライン学習 AI は現在の相手に動的に対応するように設計されている．例えばSARSA 法により相手に有利な行動の強化学習を行う AI [92] や k-NN 法により相手の次攻撃行動を予測してそれに有利な行動を出力する AI [86] である．しかしそうはいっても，相手との対戦回数が限られた状況では現在の相手に動的に対応することは一般に難しい．得られる現在の相手からのデータの量が少ないという問題を除いて考えるとしても，既存のオンライン学習型 AI では学習手法により定義される各状態に対応付けられる行動が初歩的な単位行動らしきものに多くの場合限られている傾向が見られる．おそらく行動の種類の増大は素早い学習や対応の妨げになるためだと考える．

しかし格闘ゲームには初歩的な単位行動の連続による数々の連続行動群，例えばガードしてからの反撃，コンボ攻撃，後ろに下がると見せかけての急な前進からの攻撃などがあり，勝負に大きな影響を与える．これらはルールベース型では容易に実装できる行動である一方で，機械学習により初歩的な単位行動のみから適切な連続行動を獲得することはコストの高い問題である．

そのため通常のオンライン学習型 AI によりオンラインで行われる学習がその AI の挙動に与える変化は小さいものに限定されることが多い．現在の相手への対応能力はオンライン学習型 AI の長所である一方で，連続行動を学習中の行動の候補に含めないことは損失が大きい．

ルールベース型

ルールベース型は非常に一般的であり，多くの商業格闘ゲームでも AI はルールベース型であると本研究では考える．例えば現在相手が空中に居れば空中に向けて攻撃，現在相手が一定距離以上遠く離れていれば遠くに届く攻撃を出し，それ以外の場合はランダムで行動を決めるといった AI はルールベース型の一種である．FightingICE の AI 競技会の 2013 年と 2014 年の 3 C 部門の優勝 AI もルールベース型に分類される．ヒューリスティックにより得られるある強力な連続行動たちや大局的な戦略をより容易にキャラクターに実行させられるのがルールベース型の長所である．

しかし，相手の過去の行動を If-then ルールの前提条件に含めない限りは，現在の対戦相手の戦略に対して自身の行動パターンをより有利なものまたはより不利ではないものに動的に変えることができないという欠点を持つ．上述の AI 競技会でもある上位ランクのルールベース型 AI が特定の下位ランクのルールベース型 AI に対して大差で負けたことがあった．その試合ではその高ランク AI が似た状況で同じ行動をとり続け，同じパターンに繰り返し出くわして負けという結果になった．

もちろん膨大な数のヒューリスティックな If-then ルールを高度に組み合わせて現在の相手に動的に対応するルールベース型 AI を作ることも理論上は可能といえるが，格闘ゲームのような複雑なシステムにおいてそれは現実的に非常に難しい．そこで本研究ではオンライン対応の性格を持たない既存のハンドメイドなルールベース型の AI を複数用意して

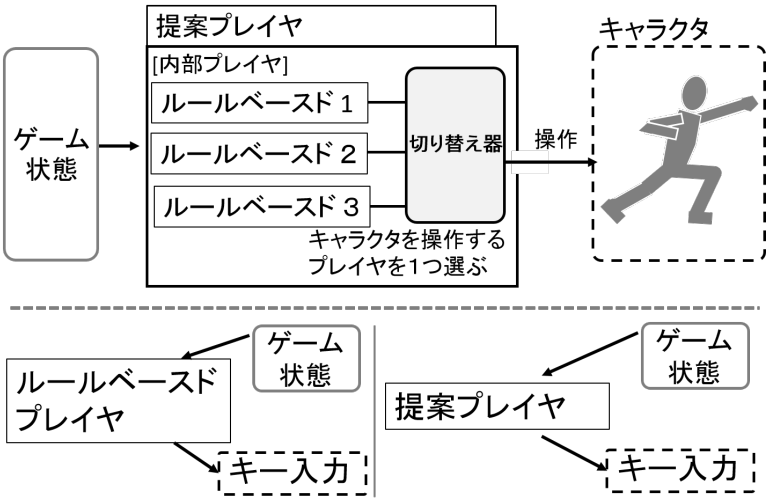

図 5.2: ルールベースド型コントローラーの切り替え

それらの切り替えを行う．それにより本研究ではルールベース型 のように連続行動を行いながらオンライン学習型 AI のように状況に動的に適応する AI を手軽に構成する手法を提案した．次章にその詳細を述べる．

5.3 適用手法

5.3.1 概観：提案プレイヤとコントローラー

本研究ではルールベース型 “コントローラー”（前節のルールベース型人工プレイヤを意味する）にオンラインな適応性を持たせるために，複数のルールベース型コントローラーを用意して，それらの間でキャラクターのコントロールの切り替えを行った．この手法は複数のコントローラーの中から現在の対戦相手に対してより効果的なものを対戦を通じて探り，そのコントローラーにキャラクターをコントロールさせる．全体的な概念図を図 5.2 に示す．つまり本研究のコントローラーは全体として複数の既存のコントローラーから成るが，混乱を避けるために，この複数の既存コントローラーを「内部コントローラー」と呼び，その全体的な形としての本研究のコントローラーを「提案プレイヤ」と呼ぶことにする．

提案プレイヤはルールベース型コントローラーの切り替えにより，敵に対するオンラインな適応性を持つ．内部コントローラーのそれぞれが，ゲームの状況を受け取ってキャラクターのキー入力を決定する能力を各々持つが，切り替え器はたった 1 つのコントローラーのみを選択する．

本研究の手法は Simon らの Multi Agent System [89] に似ており，複数のコントローラーまたはエージェントから構成される 1 つのコントローラーという点で共通している．しかしいくつかの点で異なる．

まずMulti Agent Systemは戦いの部分的な目的，例えばコンボアタック入力用，コンボアタック回避用など，に特化したエージェントを用意するが，本研究の提案手法で用意する内部コントローラーはそれぞれが戦いの全ての局面で効果的に動作するように設計されている独立したコントローラーである．

さらに，Multi Agent Systemではキャラクターをコントロールさせるエージェントの切り替えはゲームの状況がしかるべきものに一致したときに行われ，切り替えられるべき次のエージェントは設計者によって事前にその状況にふさわしいと判断されたものである．それに対して提案手法では，キャラクターをコントロールさせる内部コントローラーを一定時間の経過で切り替えし，切り替え先の内部コントローラーはその戦いで対戦相手に対して有利な戦績を残しているものである．本研究の手法の全容は図5.2に示す通りだが，以下にそれぞれの詳細と本手法の長所短所を述べる．

5.3.2 内部コントローラー

本研究が本手法で用いる内部コントローラーは既存のルールベース型コントローラーである．特にFightingICE上の格闘ゲームAI競技会では過去の大会に出場した質の高いルールベース型プレイヤがソースコードと共に公開されているので，本研究ではそれを利用してコントローラー作成の労力を劇的に軽減できる．もちろん適切な既存プレイヤ達が入手できない状況下であっても異なる長所を持った自作コントローラにより本手法は実装できる．

内部コントローラー達はそれぞれ何らかの点で他のコントローラーより優れていることが望ましい，さもなければその内部コントローラーは全体に貢献を持たない．オンライン学習型のプレイヤを内部コントローラー として利用することも可能だが，本研究では２つの理由でそれを望ましいと考えない．まずオンライン学習型プレイヤは概して計算コストを多く必要とする．そのためそれらを複数並列して用いることは計算が定められた時間内に終了しないリスクを持つ．次にオンライン学習型プレイヤは終盤ほど強くなる傾向があり，時間の経過に応じて性能が変化する．それが内部コントローラーとして用いられた時に現在の相手に対して効果的かどうかを判断するのにより多くの時間を要する．

5.3.3 SW-UCBアルゴリズムによるコントローラー切り替え

本研究では提案手法による内部コントローラーの切り替えの問題を探索と知識利用のトレードオフの古典的な問題である非定常Multi-armed-bandit 問題（MAB [91]）の一種とみなす．MAB 問題ではプレイヤーは複数のアームから1つのアームを選び，そのアームに対応付けられた報酬を得る．何回もアームを選びその累積された報酬の最大化が目的となる．非定常MAB 問題ではアームに対応付けられた報酬の確率分布が時間ごとに一定ではない．

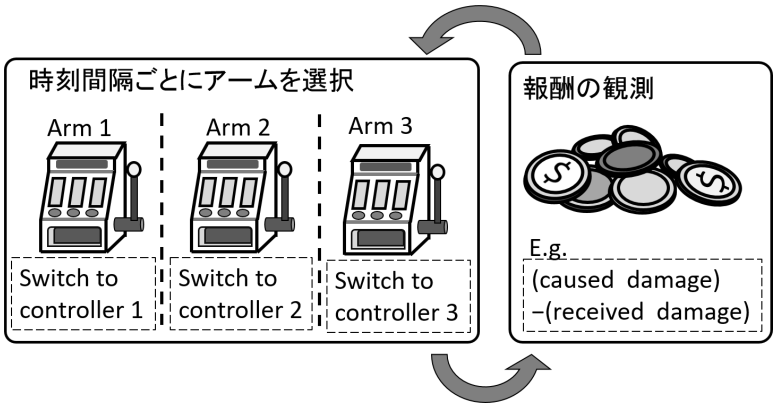

図 5.3: MAB 問題としての格闘ゲームコントローラー選択

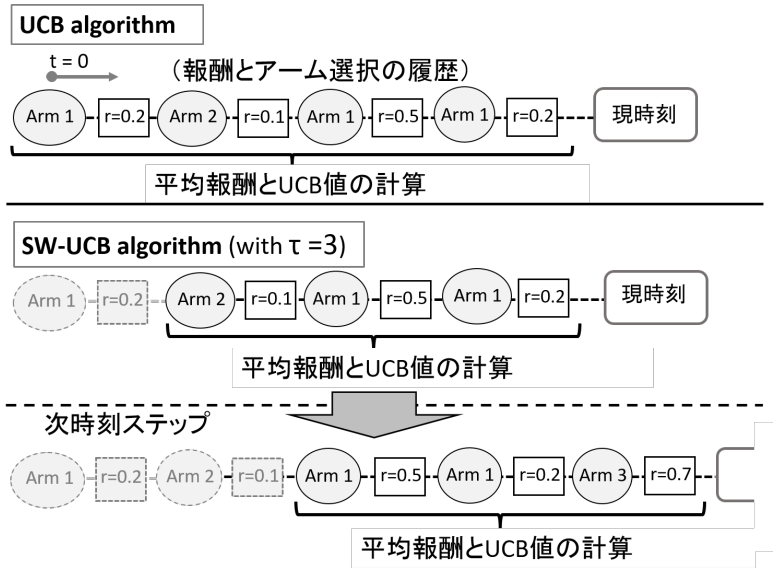

図 5.4: UCB algorithm と SW-UCB algorithm

図 5.3 に示すように，提案手法は 1 つの MAB 問題としてモデル化できる．本研究の手法では一定時間ごとにキャラクターをコントロールする内部コントローラーを複数の中から 1 つ選択する．その 1 つの内部コントローラーの選択が MAB 問題のアーム選択である．その内部コントローラーが相手にどれほど効果的に戦ったかが報酬となる．この効果性は例えば一定時間中にキャラクターが与えたダメージから受けたダメージを引いたものと定めることができる．そして本研究の状況は非定常 MAB 問題に分類される，なぜなら対戦相手の行動パターンは勝負を通じて変化しうるからである．

MAB 問題または非定常 MAB 問題について多くの研究がある．その中でも UCB アルゴリズム [91] の変種である Sliding Window UCB アルゴリズム（SW-UCB アルゴリズム）を本研究では用いる．この手法には性能について理論的なサポートが Eric らにより与えられている [90]．UCB アルゴリズムと SW-UCB アルゴリズムは次の選択するべき

アームの決定にアーム選択と利益の過去のデータを利用し，それぞれのアームの選択の指標となる値の計算方法もだいたい同じである．ただし UCB アルゴリズムは過去全てのデータを利用するが，SW-UCB アルゴリズムは過去 τ 回分のアーム選択によるデータのみを利用する．

具体的には，SW-UCB アルゴリズムでは以下に定義される $\overline{X}_t(\tau, i) + c_t(\tau, i)$ を最大化するアーム i を時間 t で選択する．

$\overline{X}_t(\tau, i)$ は平均報酬を表し，具体的な値は以下である．

$$\overline{X}_t(\tau, i) = \frac{1}{N_t(\tau, i)} \sum_{s=t-\tau+1}^{t} X_t(i)\delta(I_s i)$$

ここで，

$$N_t(\tau, i) = \sum_{s=t-\tau+1}^{t} \delta(I_s i), \qquad \delta(I_s i) = \begin{cases} 1 & I_s = i \\ 0 & I_s \neq i \end{cases}$$

さらに $X_t(i)$ はアーム i を時刻 t で選んだことによる報酬で，I_s は時刻 s で選択された行動である．

$c_t(\tau, i)$ は Exploration に対応するボーナス項であり，具体的な値は以下である．

$$c_t(\tau, i) = B\sqrt{\frac{\xi \log(t \wedge \tau)}{N_t(\tau, i)}}$$

ここで B と ξ は定数で，$t \wedge \tau$ は，t と τ のうち値の小さい方を表す．

このようにして SW-UCB アルゴリズムは今から過去 τ 回分のアーム選択の履歴を用いて，高い報酬を得られると期待されるアームを選択する．τ が無限大のとき SW-UCB アルゴリズムは UCB アルゴリズムに一致する．直近 τ 回のアーム選択よりも以前の履歴を忘れることにより，SW-UCB アルゴリズムはアームへの平均報酬値が変化する環境に対して適応する．

また本研究のように MAB 問題としてのモデル化の他には，格闘ゲームでのコントローラー切り替えを「ゲーム理論の戦略」と捉えるようなモデル化もあり得る．その場合は各時間周期ごとのコントローラー選択を戦略として考え，最適な混合戦略の導出には Martin らによる Counter Factual Regret 指標を利用した手法 [96] などが利用できる．本研究の非定常 MAB 問題モデルとの違いとしては，MAB 問題では（ゲーム理論でいうところの）純粋戦略のみの追求になる点が主に異なっている．そして，ゲーム理論によって格闘ゲームを展開型ゲームとして記述すれば非定常 MAB 問題よりもモデルが一方的にリッチになる．しかし扱う情報量は膨大になってしまい，Counter Facturl Regret を用いた最適化もオンラインで行えば敵への対応があきからに極度に遅くなることが予想されるため我々は MAB 問題によるモデル化を選ぶ．

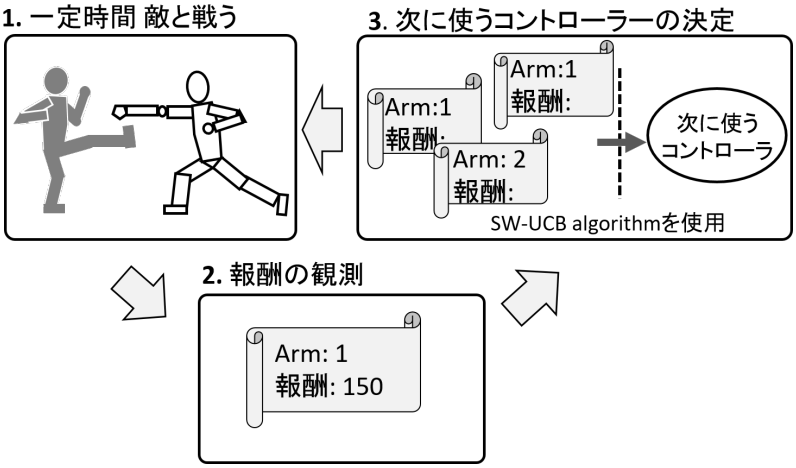

図 5.5: 提案手法の手続き

5.3.4 全体的な手続き

提案手法による全体的な手続きを図 5.5 に示す．
この手続きについて説明する．

1. 提案プレイヤが敵プレイヤと戦う．提案プレイヤ側のキャラクターは内部コントローラーの 1 つにより操作されている．

2. 一定時間戦うごとに，その時間区間内の報酬（キャラクタが与えたダメージと受けたダメージの差）を記録する．

3. SW-UCB アルゴリズムにより，次の時間区間にどの内部コントローラーがキャラクタを操るかを決定する．

5.3.5 想定される利点と欠点

利点

本研究の手法はよくあるルールベース型コントローラーに比べて現在の対戦相手にオンラインで対応することができる．対戦相手がルールベース型プレイヤなどの挙動が一貫した相手のとき，もし内部コントローラーの一つがその相手に対して有利ならばその内部コントローラーに長い時間キャラクターをコントロールさせる．また対戦相手がオンライン学習型プレイヤのようなオポネントモデリングを行う場合，そのモデリングをある程度妨げることができる．

そして本研究の手法は，ナイーブなオンライン学習型プレイヤよりも複雑な連続行動を行いやすい．この利点はゲームの状態にキャラクターの行動を対応づけるのではなく，

ゲームの各時間にある内部コントローラーの行動ルーティンを対応づけることによりもたらされる．

また内部コントローラーを別の設計者のプレイヤから流用できる場合には，本研究ではゲームシステムや各プレイヤへの特別な知識抜きにそれらのプレイヤより強いかもしれない AI を本手法により実装できる．

欠点

我々の手法は全ての内部コントローラーのもつ If-then ルールに含まれない行動をとれない．つまりオンラインの対応の細やかさに欠ける．また一定時間ごとにキャラクターをコントロールする内部コントローラーを切り替えることにより，行動の文脈を破壊する恐れがある．各内部コントローラーの設計者の想定を外れた状況に陥ることによって非常に悪い動きをある状況で行う恐れがある．

また内部コントローラーの 1 つのみを使う場合に比べて，本手法は探索（exploration）によって原理的に性能を悪くする場合がある．現在の対戦相手に対してその内部コントローラーが全ての内部コントローラーの中で勝負を通じて一貫して最も有利な場合，探索により他の内部コントローラーが使用されることで性能が劣化しうる．

こうした欠点の多くは，内部コントローラーとしてルールベースド型プレイヤを用いることが原因となるが，しかしオンライン学習型プレイヤを内部コントローラーとする場合には 5.3.3 節に示したような問題が生じる恐れがある．

5.4 予備実験

本研究ではいくつかの準備実験を実際の格闘ゲームの実験の前に行った．実際の格闘ゲーム達は複雑な環境であるが，基本的には連続する同時着手ゲームである．そのため最低限のサイズの行動群と自明な最適戦略が獲得可能な環境で，本研究の手法が適切に効果を発揮できるか確かめる．

5.4.1 じゃんけんゲーム

予備実験は複数回のじゃんけんゲームによって行う．じゃんけんゲームは同時着手ゲームで 3 つの可能行動からなる，それを act-r, act-s, act-p と呼ぶ（rock, scissors, paper）．Act-r は act-s に有利で act-p に不利である．Act-s は act-p に有利である．同じ種類の行動同士は引き分けとなる．

また以下の議論で勝率を計算するときには引き分けを 0.5 勝として勘定している．ただし本研究ではこのゲームには効果的な連続行動というものが存在しない．そのため提案手法の長所の 1 つとしての，効果的な連続行動を行いやすい，というポイントは本環境の実験からは示されないことを本研究ではここに注意しておく．

5.4.2 使用プレイヤ

本研究ではこのゲームにおける 3 つのルールベース型プレイヤとオンライン学習型プレイヤと提案手法プレイヤを用意した.

ルールベースド型プレイヤ

本研究ではルールベースド型プレイヤ，π_{win}, π_{lose} and π_{draw} を用意した．π_{win} は直前の勝負が draw でなければ，直前の相手の行動に対して有利な行動を 85%の確率で選択し，それ以外の場合にランダムに行動を選ぶ．同様に直前の勝負が draw で無いとき，π_{lose} は直前の相手の行動に対して不利な行動を，π_{draw} はその直前の相手の行動と同じ行動を 85%の確率で選択する.

これらの AI には 3 すくみ関係がある．π_{win} は π_{lose} に強く，π_{draw} に弱い．そして π_{lose} は π_{draw} に強い．例えば π_{win} と π_{lose} の対戦を考える．π_{win} が act-r を，π_{lose} が act-s を選択した場合，次の対戦で π_{win} は act-r を，π_{lose} は act-s を高い確率で選び，π_{win} は勝ちを継続しやすい．その結果として，異なるルールベースド型プレイヤ同士の対戦は片方が長期的に約 70%の割合のゲームを勝つことが実験的に解っている.

Online learning プレイヤ

本研究の用意した Online learning AI は現在の対戦の行動履歴を参照する．この AI は対戦相手の着手の系列と自分の着手の系列を以下のように保存する.

My_actions[] = {m_1, m_2, m_3, ...}

Enemy_actions[] = {e_1, e_2, e_3, ...}

ただし m_x および e_x は x 回目の対戦における自分の着手と対戦相手の着手である．ここから次の t $(t \geq 3)$ 回目の対戦における相手の着手を以下の図 5.6 の ように推測する．ただしこのアルゴリズムの中で各行動は 0，1，または 2 により数値化されて扱われているものとする.

そして推測した敵の着手に対して有利な着手をこのプレイヤは行う．このオンライン学習型プレイヤのオポネントモデリングは上述のルールベースド型プレイヤのルールの前提条件部と出力の関係を十分に捉えられる．この AI がそのルールベースド型プレイヤの 1 つと対戦すれば長期的に 90%近くの割合の試合に勝利する.

提案プレイヤ

このプレイヤは前節で述べた π_{win}，π_{lose} および π_{draw} をその内部プレイヤとして切り替える．10 戦ごとに報酬の計算を行い，次の “アーム” に切り替える．報酬は直近 10 戦の勝率である．パラメータ B と ξ は 1.0 と 10． また τ は 20 とした．実験の最初にキャラクターのコントロールを受け持つプレイヤは π_{win} とした.

1: count[] $\leftarrow \{0, 0, 0\}$
2: **for** $i = 1$ to $t - 2$ **do**
3: **if** My_actions[i]==My_actions[$t - 1$] **then**
4: **if** Enemy_actions[i]==Enemy_actions[$t - 1$] **then**
5: count[Enemy_actions[$i + 1$]]++
6: **end if**
7: **end if**
8: **end for**
9: **return** $argmax_e(count[e])$

図 5.6: 予測アルゴリズム

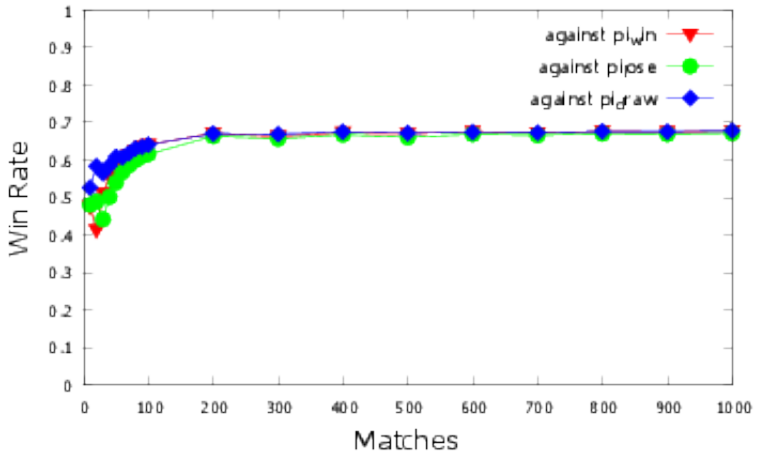

図 5.7: 提案プレイヤとルールベースド型の対戦. 1000 戦勝率を 100 試行分平均

5.4.3 実験

対ルールベース型プレイヤ

提案プレイヤは 3 種のルールベース型プレイヤと各 1000 試合戦い，これを 100 試行繰り返した．結果は図 5.7 に示す．提案プレイヤはそれぞれの対戦相手に 70%の勝率を示した．もしも内部コントローラーの切り替えを行わない場合には勝率は原理的に 50% (各相手に 70%，50%，30%ずつの平均）になるので，提案手法によるコントローラー切り替えによって単純なルールベースド型よりも高い勝率を，この場合は獲得できた．

対 オンライン学習型プレイヤ

提案手法は敵のオンライン学習型プレイヤからのモデリングをある程度妨害できると考えている．それの効果を実証するために，オンライン学習型プレイヤと 1000 回の対戦を行い，それを 100 試行繰り返して勝率を観察した．

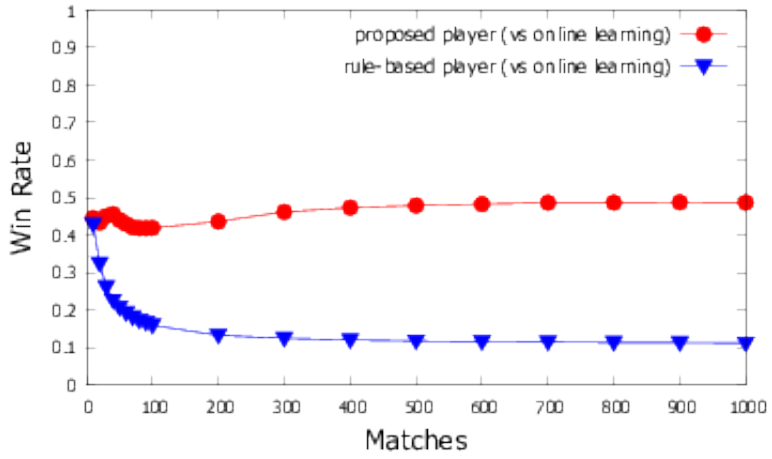

図 5.8: 提案プレイヤとルールベース型プレイヤがオンライン学習型プレイヤと対戦

結果を図 5.8 に示す．同様にオンライン学習型と対戦したルールベース型たちが 10%の勝率しか達成していない一方で提案プレイヤは 50%の勝率を達成している．よってオンライン学習型との対戦においても提案手法は，この設定においては，単純なルールベース型より高い性能を発揮したと言える．

5.4.4 結論

予備実験において提案手法はルールベース型とオンライン学習型の両方の対戦相手に対し有効に働いた．よって提案手法が，このような極度に単純化された「連続的な同時行動型のゲーム」においては効果的に働いたと本研究では結論する．

5.5 実験

本研究では提案手法が実際の格闘ゲームでよく働くかを評価するために実験を行った．

5.5.1 環境と設定

実験は FightingICE の上で行われた．提案手法プレイヤと過去のコンペティション出場 AI の対戦によって評価を行う．本研究では 3 すくみの関係をもつ 3 つのプレイヤを内部プレイヤとする提案手法 AI を構成した．その 3 つのプレイヤは 2014 年コンペティション 1C 部門の出場プレイヤでありルールベース型 である，ATTeam，T1c，Somjang AI である．大会の順位表は表 5.1 の通りになる．

表 5.1: FIGHTING ICE 2014 競技会 “1C” 部門ランキング

AI	**Ranking**	**AI**	**Ranking**
CodeMonkey	1	thunder_final	6
VS	2	ragonKing1C	7
T1c	3	ATteam	8
LittleFuzzy	4	SomJang	9
PnumaSON_AI	5	ThrowLooper	10

これら 3 つのルールベースドプレイヤは自他の行動履歴を利用せず，単に現時刻でのゲーム状態のみから次の行動を決定する．ゆえに各時刻ステップにおいて本研究の提案プレイヤは以下の手順を実行する．

1. 現在のゲーム状態をデータとして受け取る
2. もしも一定の時刻の間隔が経過していたら，SW-UCB アルゴリズムを用い内部コントローラーの切り替えを行う
3. 現在選択されているコントローラーに現在のゲーム状態データを渡し，次行動を受け取る
4. 最後に，その得られた次行動を提案プレイヤの出力としてゲームシステムに出力する

本項の実験で SW-UCB でのアームの切り替えはゲーム内 180 時刻ステップ（現実の時間で 3 秒に相当）ごとに行う．報酬は，その時間ステップ内に相手に与えたダメージと自分が受けたダメージの差に設定する．しかしいくつかの例外がある．その時間ステップ内に自分がダメージを受けなかった場合に報酬値は十分に大きな値として定め，いかなる状況でも次の時間ステップには同じアームが選択される．その状態から最初に自分がダメージを受けた時にその時間ステップの報酬は十分に小さな負の値となり，その大きな報酬値の影響をキャンセルする．また SW-UCB 手法のパラメータとして B は 100，ξ は 0.5，τ は 16 とした．

実験対戦相手としてのプレイヤは，ルールベース型の ATTeam，T1c，Somjang AI である．また，オンライン学習型プレイヤとして競技会 1 位の CodeMonke を用意した．

評価指標として FightingICE プラットフォーム内のスコアシステムを用いた．各試合は 3 つのラウンドで構成され，各ラウンド後に以下の値のスコアを受け取る．

$Score = \frac{opponentHP}{selfHP+opponentHP} * 1000$

このスコア値の最小値は 0 で最大値は 3000 である．そして両者が戦いで互角だった場合にはスコアは 1500 となる．

表 5.2: 対ルールベースド型プレイヤ 100 戦平均スコア

	opponent				
	Switch AI	ATTeam	Somjang	T1c	Total (95% CI)
Switch AI	-	2172	1639	1577	**5388** (±76)
ATTeam	828	-	397	2647	3872 (±96)
Somjang AI	1361	2603	-	1019	4983 (±93)
T1c	1423	353	1981	-	3757 (±76)

表 5.3: 対オンライン学習型プレイヤ 100 戦平均スコア（一部 300 戦）

	CodeMonkey(95% CI)
Switch AI (300 matches)	579 (±58)
ATTeam	312 (±54)
Somjang AI	316 (±56)
T1c (300 matches)	552 (±63)

5.5.2 結果と考察

本研究では結果を表 5.2 と表 5.3 に示す．“95% IC” は 95%信頼区間である．

提案プレイヤは有意に他のルールベースド型プレイヤを上回った．一方で提案プレイヤはオンライン学習型プレイヤに対し，他のルールベースド型よりわずかに高いスコアを取りつつも，それほど効果的でなかったようにみえる．

T1c (表 5.2) との試合においては提案プレイヤは特に低いスコアを記録した．これは ATteam のコードに記述された特定の連続行動のためである．この連続行動は T1c にとても有効に働くものの，タイミングがシビアであり，本研究の「一定時間毎の切り替え」によるシステムではうまく効果を発揮しなかったと考える．学習履歴の例を図 5.9 と 5.10 に示す．これらの図で 1, 2, 3 と y 軸上に記されたのはそれぞれ人工プレイヤの Somjang AI と T1c と ATTeam を意味し，提案プレイヤ内部で各時間にどのプレイヤに操作が切り替えられたのかの履歴である．図 5.9 は適応がうまくいったケースで提案手法は相手に有利なコントローラーに頻繁にキャラクタを操作させている．一方で図 5.10 の場合では適応がうまくいってない．ATTeam と T1c は，敵プレイヤ T1c に対して有利なコントローラーではないにも関わらず頻繁に選ばれている．しかしそれでも，敵 T1c に対して不利となる ATTeam はゲームが終盤に進むにつれて次第に選択されにくくなっている．

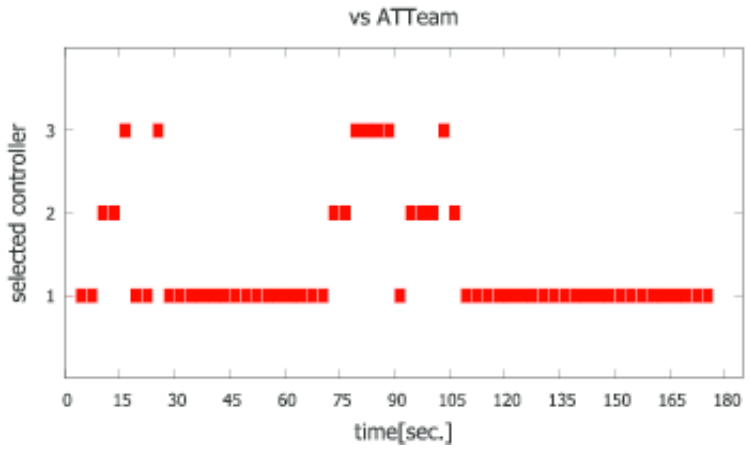

図 5.9: 対 ATTeam 学習履歴 (y 軸 1:Somjang AI, 2:T1c, 3:ATTeam)

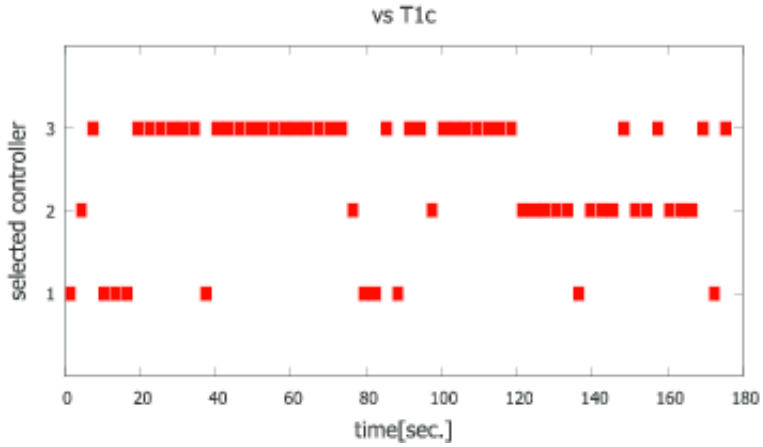

図 5.10: 対 T1C 学習履歴 (y 軸 1:Somjang AI, 2:T1c, 3:ATTeam)

5.5.3 パラメータ変更による検証

本研究ではパラメータを変えて実験を行った．変更したパラメータは，コントローラー切り替えの時刻間隔と τ である．τ は行動履歴データをいくつまでさかのぼって利用するかのパラメータであり，もしこれが無限題なら SW-UCB アルゴリズムは単なる UCB アルゴリズムと等価となる．

τ が小さいほど切り替えは “近視眼的” になって，ごく最近の「各コントローラーが得た報酬値」のみしか考慮しなくなる．よって小さい τ はオンライン学習型プレイヤと戦うときに，相手の挙動の変化に素早く適応でき，有効であると考える．似たような理由で，コントローラー切り替えの時刻間隔を短くするとまたオンライン学習型に有利になると予想できる．しかし逆に，小さい τ や切り替え時刻間隔は提案システムの挙動の安定性を損なうリスクがある．

表 5.4: τ を変化させた場合の勝率変化

	opponent				
	ATTeam	Somjang AI	T1c	CodeMonkey	Total(95% CI)
$\tau = 4$	1862	1629	1111	664	5266 (±146)
$\tau = 16$	2172	1639	1577	573	5961 (±96)
$\tau = 32$	2085	1591	1186	682	5544 (±138)
$\tau = \infty$	2359	1613	1401	557	5930 (±154)

表 5.5: コントローラー切り替え間隔を変化させた場合の勝率変化

	opponent				
intervals	ATTeam	Somjang AI	T1c	CodeMonkey	Total (95% CI)
60	2359	1546	1093	760	5758 (±134)
90	1781	1564	1361	753	5459 (±120)
180	2172	1639	1577	573	**5961** (±96)
600	2019	1540	1438	442	5439 (±127)
1800	2045	1554	1362	406	5367 (±101)

このパラメータ変化の実験も，相手プレイヤは前節と同じく，ATTeam，Somjong AI，T1c，CodeMonkey を用意した．100 戦の平均スコアを用いて評価した．変更対象でないパラメータは前節同様である．τ を変化させるときはコントローラー切り替え時刻間隔を 180 に，切り替え時刻間隔を変化させるときは τ は 16 に固定した．

結果を表 5.4 と 5.5 に示す．理論的な予測，つまり「τ が小さいほどオンライン学習型に対し有利でルールベースド型に不利」という傾向とは一致しない結果が得られた．しかし表のスコアが最小で 0 から最大 12000 までの分布であることと結果の数値を合わせて考えると，提案手法がパラメータの変化にある程度頑健である，という解釈の仕方も可能である．

5.6 結論

本研究ではオンライン学習型プレイヤのような適応とルールベース型プレイヤのような効果的な連続行動の，双方の利点を併せ持つ人工プレイヤの開発手法を提案した．さらにその提案手法の性能を，研究において代表的な格闘ゲーム環境を用いて評価した．提案手法は 3 種の既存ルールベース型プレイヤを切り替えて使うことで，そのそれぞれよりも

高い勝率を発揮し，オンライン学習型プレイヤに対する性能もわずかに向上した．

提案プレイヤは，単に現時点における強さの点では既存の大会優勝プレイヤを上回っていないが，本手法は「既存のルールベース型プレイヤ」を組み合わせる枠組みであるため利用可能なルールベース型プレイヤの性能に応じてより高い性能を発揮できる見込みがある．また本手法の利点として，一般に人間プレイヤから挙動パターンを読まれてしまいやすいルールベース型プレイヤを組み合わせることでパターンを読まれにくくする．そのため人間プレイヤの対戦相手としての適用を考える場合も，飽きにくくなり「遊んで楽しい」人工プレイヤの実現に本手法は役立つと考えられる．

第6章　人間らしい弾避けを行うシューティングゲーム人工プレイヤ

本章は会議への投稿論文を元に書き直したものである [40]．本章では，多様な対象と目的のゲーム人工プレイヤのための，リアルタイム制ゲームにおける強さ以外の目的の人工プレイヤ作成に着目した研究について述べる．この「リアルタイム制ゲーム・強さ以外の目的」の領域の中で，幅広い未解決な課題の中から本研究が対象に選ぶのは，シューティングゲームにおける人間らしい挙動の人工プレイヤ開発である．

6.1　はじめに

ゲームの人工プレイヤに関する研究の一環として，単に競技に強いプレイヤの研究だけでなく，近年は遊んで楽しめる人工プレイヤや動作が人間らしい人工プレイヤに関する技術が注目を集めている．人間らしい人工プレイヤの実現は例えば人間の認知過程の究明や一緒に遊ぶ人間プレイヤへのゲーム体験の向上を目的として行われ，その適用対象も幅広い．古典的なターン制のボードゲームにとどまらず，現代的なリアルタイム制のビデオゲームも広く対象ジャンルとして研究されている [97] [98]．

しかしビデオゲームの一ジャンルであるシューティングゲームでは著者の知る限り人工プレイヤに関する研究が非常に少ない．シューティングは日本では昔からかなりメジャーなジャンルであり，また他のアクションゲームとは異なる性質を備えている．特に対戦型シューティングというサブジャンルでは，相手を務める人工プレイヤの挙動が人間プレイヤの満足度に直結するため，シューティングにおける人工プレイヤ研究は重要であると考える．

そこで本研究では，シューティングゲームで人間らしい人工プレイヤを実現するための部分問題として，人間らしい弾避けを行うプレイヤの開発を試みる．環境は,『東方』シリーズ [99] をモデルとしたシューティングのオープンソース [100] を参考にした自作環境を用い，手法は経路探索と Influence Map による危険度判定を併用する．そしてその提案型の人工プレイヤがどれほど人間らしく見えるかについて被験者実験により評価する．

6.2 背景

6.2.1 人間らしいゲームプレイヤ

ゲーム人工プレイヤの挙動の人間らしさについては将棋，囲碁，ロールプレイング，一人称視点シューター（以下 FPS）やアクションなど様々なジャンルで研究が行われている．古典的ボードゲームでは，将棋では特定のプレイヤの棋風再現の研究 [101] が見られ，囲碁では人間らしい自然な着手による手加減が既存研究 [102] により追求されている．またターン制のビデオゲームジャンルとしてロールプレイングでは人間の認知モデルを模倣した AI プレイヤにより，一緒に遊ぶ満足度を上げる試みがされている [103]．

対して近代的なリアルタイムのビデオゲームでは，アクションや FPS は研究が活発なジャンルの例である．アクションでは『Super Mario Bros.』[104] シリーズをモデルにした環境を用いた，人間らしい人工プレイヤの競技会が開かれている [97]．2012 年の競技会では，ニューラルネット，influence map，nearest neighbor 法などが参加人工プレイヤの設計に用いられている [105]．また，人間の不完全な認知様式を模倣したエージェントによる強化学習も試みられている [106]．

また FPS でも人間らしさを競う人工プレイヤ競技会が開かれていて [98]，Finite State Machine の適用 AI [107] や，Behavior Tree と Neuro Evolution 手法を組み合わせた AI [108] などが報告されており，それぞれ競技会で上位にランクインしている．

競技会に関連しない研究例としては，FPS の『Quake』にて人間プレイログの模倣を Imitation Learning により行った人工プレイヤの開発 [109] や，FPS『Unreal Tournament 2004』にてリカレントニューラルネットワークを用いた人工プレイヤの学習 [110] が報告されている．

6.2.2 シューティングゲーム

シューティングゲームは日本で人気の根強いジャンルであり,『グラディウス』[111] や『ゼビウス』[112] などが有名なタイトルである．主な形式としてプレイヤの操るキャラクタが弾・敵キャラ・(地勢等）障害物を避けながらステージのゴール到達または敵のボスキャラクタの破壊を目指す．図 6.1 に一例を示す．

他のリアルタイムゲームとの違いを整理する．Super Mario などのアクションゲームと違ってシューティングは自動的に画面がスクロールしていく形式がほとんどなので，能動的にキャラクタをゴールに向けて移動させていく必要がない．そのため人工プレイヤ実装の面ではアクションに比べて「ゴールへ向けた経路計画」を行わずともクリアできるが，その分，弾や敵の回避動作により細かい動きが要求されやすく，その動作の様子が人間らしさの印象に大きな影響を与えると考えられる．

また FPS と比べれば，アクションと同様の「画面のスクロール」に関する違いがある．FPS は敵の殲滅などの目的を目指してキャラクタを能動的に目的地に移動させなければならない．さらに FPS では敵からの弾は，目視してから適切に回避することは極めて難

図 6.1: シューティングゲームの一例.『龍神録の館』[100] を参考に作成した環境.

しい．よって FPS ではそもそも敵から弾を発射されないように有利な位置取りを行うことや襲撃のタイミングを計画することが重要視される事が多い．一方シューティングでは敵に弾を撃たれること自体をそこまで防ごうとする必要はないが，回避可能な速度の弾をどう避けるが重要になる．

よってシューティングプレイヤへの適用に際しては，FPS やアクションを専ら題材とした既存研究手法と同様の手法や工夫が効果的である保証がなく，本研究ではシューティングゲーム固有の特徴に基づいてアプローチを定めたい．シューティングの人工プレイヤに関する先行研究は少ないが，複雑な弾の回避を行うために A*法による経路探索と Influence Map を組み合わせた設計が行われている [113]．この研究における手法設計は後述する本研究の人工プレイヤとかなり近い．しかし本研究の解釈する限り，Influence Map 法の値の割り当てに弾の通過の情報を使うか速度を使うかの違いや，経路探索に関する諸工夫，また本稿が『人間らしさ』を目的にして実装と評価を行っている点などに相違がある．

6.2.3 既存シューティング用プレイヤに見られる問題点

シューティングにはサブジャンルとして対戦型シューティングがあり，人間の対戦相手を人工プレイヤが務めることがある．特に既存の対戦シューティングでは人工プレイヤ動作の不自然さによって人間プレイヤの楽しさに問題を与えかねない事例が見られる[1]．そのため本研究では対戦型シューティングで人間らしい人工プレイヤを作りたい．

この対戦型シューティングはシューティングの拡張型であり，

- ときどき相手プレイヤに妨害用の効果（弾や障害物の増加）を与えることができる
- 相手の一定回数以上のミス（弾や障害物への衝突）が勝利条件となる

[1]『東方花映塚』[114] では人工プレイヤの強さは一定時間の経過によって急激に低下するように設計されている．よって一定時間が経過するまではどれほど強い妨害を与えてもミスをせず，一定時間経過後にはかなり弱い妨害を与えてもミスをするため，人間プレイヤ側の妨害攻撃のし甲斐が損なわれると本研究では考える．

という 2 点を除けば，各プレイヤにとっては 1 人用シューティングとほぼ等価である．そのためまずは 1 人用シューティングで人間らしいプレイヤを作ることには価値がある．

対戦型シューティングのタイトルで本研究が人工プレイヤの動作を詳細に観察できたのは，著名な同人ソフトの『東方花映塚』[114] のみである．よって，本節はたった 1 つのタイトルを基に対戦/シューティングの人工プレイヤ全般の問題点を述べることになる．しかし，およそどのようなシューティングのタイトルでも（もし生じれば）共通して問題となると考える点，なおかつその十分な解決策があきらかに既知ではないと考える点に絞って以下に論じる．

素早く精密な回避

観察した人工プレイヤは密集した弾を非常に緻密な回避動作で，しかも迷いなく素早くくぐり抜けることがある．数ピクセル単位での当たり判定の回避を行うこともあり，このような緻密な回避は Infinite Mario においても人間らしくないと指摘されている [106]．これは人間にとって難しい動作なため機械的に映ると考える．もちろん人間プレイヤであっても上級者は極めて精密な動きをすることはあるが，そういったプレイヤは今回人間らしさの対象として想定しない．

局所的視野

また前節のような緻密な動作を，必要に迫られたときのみならず，少し遠くに安全に弾を潜り抜けられる場所があっても行うときがある．人間はこれとは逆に，現在位置にとどまって難しい弾避けをするよりも遠回りして安全な弾避けを好む傾向がある．そのためこうした局所的な視野にしか注目しないような人工プレイヤの行動判断は非人間的な印象に結びつくと考える．

細かい振動

AI は，人間にとって到底可能ではない程の短時間でのキー入力の切り替えを行うことがある．これがゲーム画面上の動きとしてどの程度不自然な様子に映るかはタイトルにより程度の差はあるが，概して機械的な印象に結びつくと本研究では考えている．予備実験では人工プレイヤの「1 フレームのみの移動キー入力」の頻度は人間の約 60 倍であることが分かった．

6.3 接近法

前項の問題点を受け，本研究は「シューティングにおける人間らしい動き」を，主に人間中級者をモデルとして以下のように想定する．

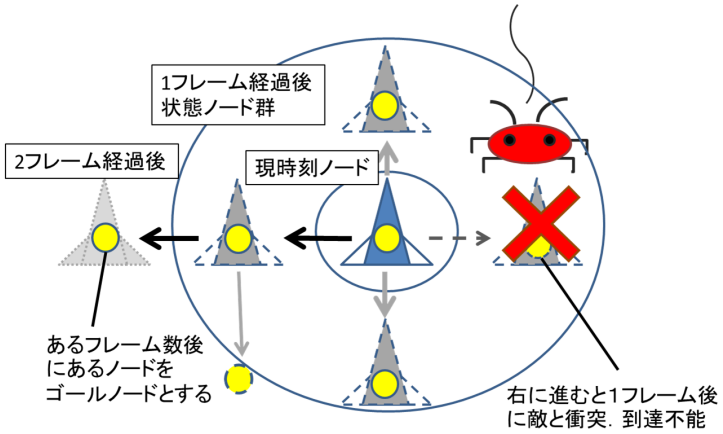

図 6.2: シューティングゲームにおける経路探索．数フレーム後に（被弾せず）到達できる場所を列挙してそれぞれ評価する．

- 弾や敵をなるべく余裕を持って回避する
- キー入力の高速かつ頻繁な入れ替えを行わない

このような動きを実現するため，経路探索をベースにしたキャラクタ移動法を提案する．この手法の適用対象として想定するのは，等速直線運動する弾を移動によって回避するのみの簡単なシューティングゲームである．ショットやボム，グレイズ（弾かすりによる得点ボーナス），あるいはより複雑なルールの影響は現段階で考慮していない．しかしそうしたルールがある場合にも，移動のみによる人間らしい回避運動は必須であると考える．

6.3.1 経路探索

提案手法は十数フレーム先までの取りえる経路を調べ上げ，最も危険度の少ない経路の初手を選ぶ．模式図を図 6.2 に示す．キャラクタの移動または無移動を枝として，フレーム経過したゲーム状態をノードとする．被弾してしまうゲーム状態は到達不能として，探索開始ノードから一定フレーム後経過したノードを経路のゴールとする．

各経路に割り当てる危険度の算出には次節の Influence Map による被弾の危険度を用いる．その経路が通過する全てのノードの危険度を重み付け合計して経路の危険度とする．詳細は付録に記すが，開始ノードから時間経過が少ないノードほど重みが低い．これは「安全な場所を通って危険な状態に飛び込む」移動より「少し危険な道を通って安全な場所に出る」移動の方が，人間らしさ・生存能力の高さの両方の観点で，優遇されるべきと考えての措置である．

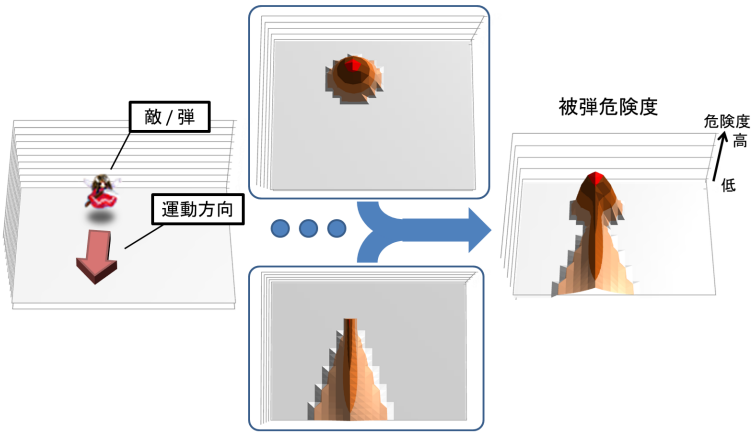

図 6.3: シューティングゲームにおける被弾危険度生成．単体の弾または敵に対して，その周辺（円形のエリア）と将来軌道の周辺（三角形のエリア）に高い危険度を割り当てる．

6.3.2 Influence Map によるノード評価値

ノード状態評価には “被弾危険度” の Influence Map を用いた．弾を余裕をもって避ける，そして弾の射線を嫌う動きを再現するため，弾の周辺や弾の射線付近に高い値を割り当てるような被弾危険度の関数を設計し，そのうえでそれを組み合わせ，キャラクタ位置の危険度を計算する．

被弾危険度

ある 1 つの弾または敵はその周辺に，図 6.3 に示すような景観の被弾危険度を割り当てる．人間にとってのある程度の観測の不確実性を織り込みながら，円形の範囲はその弾や敵の現在の位置に対する被弾のリスクを表し，三角形の範囲は近い将来の軌道からの被弾のリスクに対応する．

そして画面上の各点について複数の的や弾からの被弾危険度がある場合はその最大値を割り当てる．その概略を図 6.4 に示す．被弾危険度算出のための詳細な関数の設計は付録に譲るが，このような分布の被弾危険度によって AI プレイヤは

- 弾や敵に（被弾しない場合でも）近づきすぎない
- 弾や敵が将来通りそうな場所から遠ざかる

ような動きを行うとことを期待する．そのため 6.2.3 節と 6.2.3 節に挙げた挙動の問題の発生を抑制できると考える．

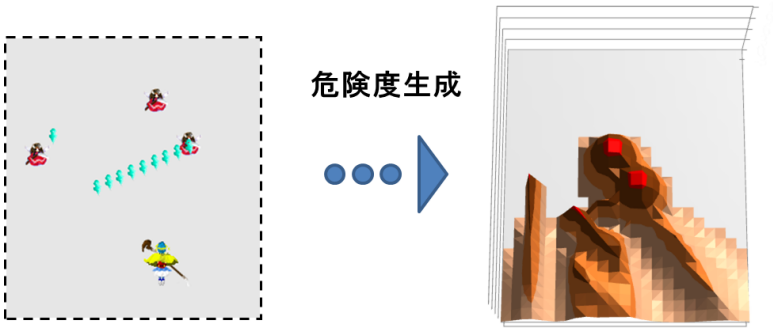

図 6.4: シューティングゲームにおける複数の敵や弾からの被弾危険度生成．図の右側の赤い敵 2 体は右下へ，左側の敵 1 体は右に運動．各敵や弾からの危険度を Max 演算したものを各点に割り当てる．

Influence Map

本研究では計算時間の都合から，ゲーム画面を等間隔なグリッド状に分割してそれぞれに被弾危険度を割り当てた．このように地勢をグリッド分割して，各物体から各グリッドへの何らかの影響の度合いを重ね合わせて割り当てる技術は Influence Map と呼ばれ，リアルタイムシミュレーションゲームの人工プレイヤ設計などに利用されたり [115]，シューティング A で短い探索時間でも危険を予期できるようにする目的で使われた例がある [113]．類似の技術としてロボティクス分野の Potential Field 法 [116] があるが，エージェントの経路決定までが手法に含まれるか，複数種類の値をグリッドが保持し得るかの相違点がある．

そしてキャラクタ位置に対する評価値には，隣接する複数のグリッドの被弾危険度を補間した値を用いた．その補間の仕方や評価値に関する正確な記述は付録に譲る．

6.3.3 その他の諸工夫

また他にも本研究では探索に工夫を加えた．

複数フレームにまたがる移動の強制

6.2.3 節と 6.2.3 節に挙げた問題点に注目する．シューティングのようなリアルタイムゲームでは計算時間の短さから探索が浅くなりがちで，そのため画面を大域的に見た時の安全な領域を探索に含めることが困難であると予想される．

そこで本研究では図 6.5 に示すように，探索木の 1 行動をキャラクタの複数フレームにまたがる一方向への移動に対応付ける．これによって探索は同じ深さでもより遠くの将来を予見することができ，人工プレイヤの行動がより大域的な視野からの判断に基づくもの

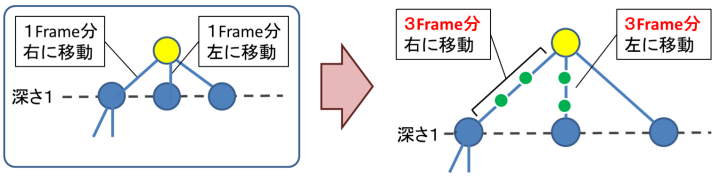

図 6.5: 複数フレームにまたがる移動を 1 つの枝とするグラフ

になると期待できる．また同時に，人工プレイヤが出力する移動も探索の通り複数フレームにまたがる一方向移動とすれば細かい振動の動きも抑えられると考える．

ただしこの工夫にはリスクもあり，一定フレームずつの規則的な行動が機械的な印象を与えたり，または取れる行動が著しく制限される（例えば 10 フレームの移動を強制した結果，人間にとって簡単に避けられる弾が人工プレイヤに避けられなくなる等）ことで人間らしさが損なわれる可能性もある．

反射神経を模した障害物制御

また本研究では人間の反射神経を考慮して 0.4 秒以内に生成された敵や弾を存在しないものとして探索した．これによって，「画面端から敵が現れた時、 1 フレーム以内に敵の軌道から離れ始める」または「自分を狙って弾が発射された瞬間にその射線を避ける」ような不自然な動きが抑制できると考える．このような人間の反射神経を模した人間らしさへの接近法は，藤井らの手法 [106] や，FightingICE プラットフォーム [117] のシステムにもみられる．

6.4 評価

本研究では提案した手法の人間らしさの度合いを検証するために，提案手法 AI を実装し，被験者実験でチューリングテストを行った．

6.4.1 使用環境

本研究では図 6.6 のような環境を作成した．これは商業タイトル『東方シリーズ』[99] をモデルに作られたシューティングゲームのオープンソース [100] を参考に，C#にて必要な機能の付け足しや削除を行なったものである．

素材画像および画面領域の幅，キャラクタの速度や当たり判定のサイズなどは全て元のオープンソースの素材や数値を流用している．オープンソース版との主な差分として，使用環境には低速移動とボムとアイテムが無い．一方でキャラクタの移動を AI プレイヤが行えるようにしている．また，キャラクタ画面は 2 画面分に増えているが，これは将来

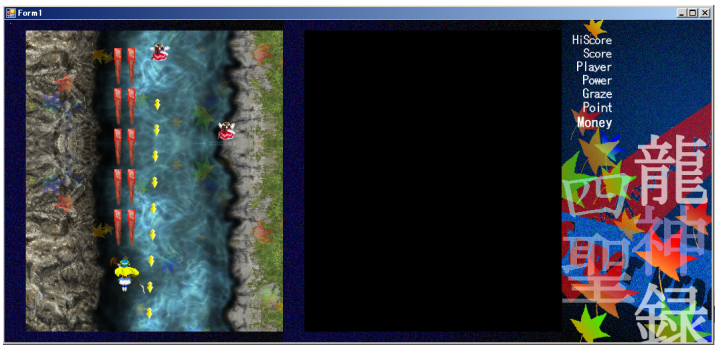

図 6.6: 使用環境（図 6.1 再掲）のスクリーンショット.『龍神録の館』[100] を参考に作成し，対戦型シューティングへの拡張を考えて 2 画面分のプレイヤ領域が用意されている．

に対象問題を対戦型シューティングに拡張することを考えての設計である．現時点では 1 画面分しか使われない．

つまり，対戦型を将来的に想定した環境を用いて 1 人用シューティングの動画を被験者に見せている．本研究は最終的に対戦型シューティングで人間らしい振る舞いをするプレイヤの構成を目指すが，本稿では対戦でなく 1 人用シューティングのプレイヤ挙動の自然さを評価していることに注意されたい．

6.4.2 被験者実験

この環境で設計した人工プレイヤの人間らしさを確かめるため被験者実験を行った．この実験では様々な被験者にゲームプレイの動画を 2 種類見せて，どちらがどれほど人間らしかったかのみ評価してもらう．

本研究の目的からすれば「弾を余裕を持って避けているか」「キーの細かい入れ替えを行っていないか」といった評価項目の導入も検討できるが，その質問が被験者に先入観を与えてしまい「人間らしさ」の評価に影響するリスク（例えば，直感的に人間らしくない印象の動きのキャラクタを見ても単に弾を余裕を持って避けているだけで，人間らしさを高く評価してしまう等）を考えて，導入しなかった．そのかわり,「弾を余裕を持って避けているか」および「キーの細かい入れ替えがあるか」の評価に関してはアンケートの自由記述欄のコメントからその達成具合を推し量る．

使用プレイヤ

動画でプレイを比較してもらうためのプレイヤとして 3 種の人工プレイヤ，**Base 探索**，**I-Map**，**I-Map+複数 F** を用意した．各プレイヤに搭載される機能は表 6.1 の通りで，**Base 探索**プレイヤは被弾につながる行動のみを回避する．

表 6.1: 実験使用プレイヤのオプション．経路探索，認知遅れを模した障害物制御，Influence Map，複数フレームにまたがる移動の制御，それぞれの搭載/非搭載．

	経路探索	認知遅れ	I-Map	複数フレーム
Base	○	○	×	×
I-Map	○	○	○	×
I-Map+複数 F	○	○	○	○

本稿で採用した工夫は多いが，本研究が特に大きな工夫と考えるもののみに焦点をあてて各プレイヤを設定している．また各人工プレイヤの持つパラメータの設定などは付録に記してある．

またこれらの人工プレイヤに加えて，シューティングの熟練人間プレイヤと初心者人間プレイヤも比較の対象に加えた．

実験条件

被験者の人数は 30 人で，シューティングゲームの経験は初心者（タイトル 1 本も遊んだことなし）が 4 人，中級者（1 から 5 本経験）が 16 人，熟練者（タイトル 6 本以上経験）が 10 人参加した．初心者のうちビデオゲームそのものを遊んだことがない被験者は 1 人のみだった．性別は男性が 26 人，女性が 4 人である．大学院生が中心で 20 歳以上 30 歳未満の被験者が 28 名，残りは 31 歳以上 40 歳未満である．

手順は以下のようにした．まず被験者は対象ゲームを 3 分間遊んでゲームに慣れる．その後，このゲームをプレイするプレイヤまたは人間プレイヤ（計 5 体）の約 30 秒の動画を 2 つずつ 8 セットを指定された順で見た．各動画，そして被験者がプレイする時のゲームでは敵と弾の配置が同一ではない．しかし，ある 2 体の敵の弾の発射パターンを入れ替えるだけに留めるなど，違いがあまり大きくならないよう配慮した．そして動画内の敵と弾配置は生存の難度がかなり簡単なレベルに設定してある．

そしてそれぞれの人間らしさを，単独で（絶対評価），さらにはもう片方と比べて（相対評価）の 5 段階評価でスコア付けした．

見せる動画の種類と順序を以下に示す．計 16 個の動画を使用している．被験者は半数ずつ 2 グループに分かれ，各セット内で 2 つの動画を逆順に見た．各プレイヤは最低 3 回以上比較の対象になるようセットは設計されている．その 3 回では前段落で述べた通りわずかづつ弾と敵配置が違う動画が使われるが，各セット内で全ての被験者が見る動画は同一のものである．

1 セット目 Base 探索と I-Map

2 セット目 I-Map と I-Map+複数 F

3 セット目 Base 探索と人間上級者

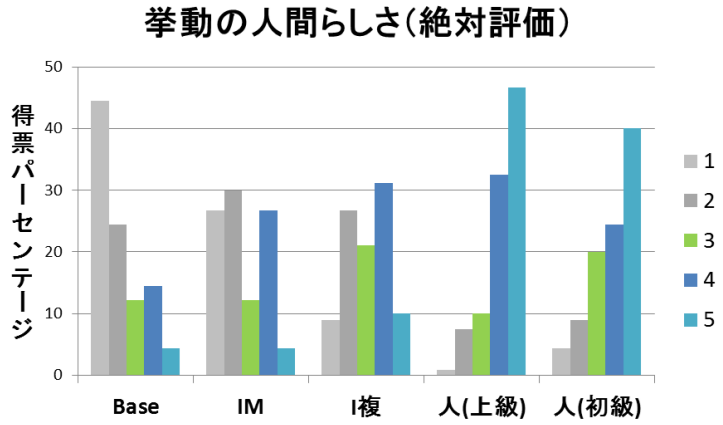

図 6.7: 絶対評価による人間らしさの獲得スコア．評価「1」は『人間らしくない』,「5」は『人間らしい』．グラフの縦軸は，各評価獲得数の全得票数に占める百分率．

表 6.2: 相対評価による人間らしさの獲得スコア平均値と 95 %信頼区間．列のプレイヤと比べて行のプレイヤが人間らしく感じられた度合い

	Base	IM	I 複	人 (上)	人 (初)
Base		2.6 (±0.52)		1.5 (±0.27)	2.1 (±0.40)
IM	3.4		2.0 (±0.40)	1.8 (±0.31)	
I 複		4.0		2.2 (±0.39)	1.8 (±0.28)
人 (上)	4.5	4.2	3.8		2.4 (±0.33)
人 (初)	3.9		4.2	3.6	

４セット目 **Base** 探索と人間初級者

５セット目 **I-Map**+複数 **F** と人間上級者

６セット目 **I-Map**+複数 **F** と人間初級者

７セット目 **I-Map** と人間上級者

８セット目 人間上級者と人間初級者

結果

結果の平均スコアとヒストグラムを表 6.2，6.3 および図 6.7 に示す．

相対評価を見ると，Base 探索より I-Map の方がより人間らしいと評価され，そして I-Map+複 F の方がより人間らしいと評価されている傾向がみえた．そかしこれら手法はまだ人間プレイヤからは人間らしさに欠けていると評価された．表 6.3 に示す絶対評価も，そうしたプレイヤ間の人間らしさの順序関係を反映している．

表 6.3: 絶対評価による人間らしさの獲得スコア平均値および 95 %信頼区間

Base	IM	I 複	人 (上)	人 (初)
2.1 (±0.26)	2.5 (±0.26)	3.1 (±0.24)	4.2 (±0.17)	3.9 (±0.24)

考察

アンケートの自由記述では，Base 探索について，人間らしくない理由として「ギリギリまで避けない」「狭いスキマを抜ける」といった記述が複数見受けられた．これらの理由が Base 探索のスコアの低さに貢献したと考えられる．しかし少数だが「これは（人間の）上級プレイヤかもしれない」という理由で低くないスコアをつける例もあった．

対して，I-Map については「密度が低いところを探して動く」,「大回りして避けている」という視野の大局性に関する人間らしさへの理由が見られた．ただし I-Map にはそうした人間らしさの理由の 2 倍以上の量,「普段からカクカクしている」「不必要な小さい動きが多い」という振動的な動きへの記述も寄せられた．

I-Map+複数 F に関しても同様に，視野の大局性に関する記述がされた．それと同時に「無駄な動きが多い」「落ち着きがなくせわしない」という，絶えず動きがちな点へのコメントもされた．

人間上級者に関しては,「滑らか」「大局的に避けている」という具体的な記述や,「自分の予想通りに動いた」「違和感がなかった」という漠然とした記述も見られた．本研究が想定していなかった人間らしさの理由としては,「1 度避け始めた方向に（そのまま）避けたがる」という判断の一貫性に関するもの,「目に見える隙間だけを通っている」という人間の視覚的認知に関するものがあった．人間らしくなさの理由としては,「無駄が無さ過ぎる」「画面端を怖がらなさすぎる」という熟練度の高さが仇になったものが見られた．

人間の初級者は人間らしさに関する記述は「ミス」に関するものが複数見られた．このプレイヤのみプレイ中に被弾していて（6 セット目に 1 度のみ），それを人間らしいと指摘する回答者が何人かいた．だが一方でその不慣れさが原因で，人間らしくないと指摘されることも多かった．例えば「斜め移動を使わない」「横にばかり動く」という，動作の種類の少なさに違和感を覚えたコメントが 6 人から指摘された．

これら自由記述の結果は客観的データとしての信頼性には欠けるものの，おおむね本研究の想定する「シューティングにおける人間らしい動き」の要因である「余裕を持った回避」と「細かすぎるキー入れ替えの不在」の妥当性を支持していると考えられる．それと同時に提案手法によってその「人間らしい動き」の傾向に近づいていることも見て取れた．

本研究の手法の有効性について考えると，自由記述から Influence Map による大局的な弾の回避は人間らしさの印象に貢献したと解釈できる．一方で複数フレームの移動による微細な振動の抑制については，自由記述からはその効果を強く示唆するようなコメントを発見できなかったが，評価点の改善により効果はあったと判断できる．

対して，現在の設計の不十分な点として動作切り替え頻度の調整が考えられる．Influence Map 使用の AI 2 種に対して「動きに落ち着きがない」という指摘が目立ったが，大量の

弾が発生したとき移動キーを入力する時間の割合が高くなることは人間プレイヤにもよくある．しかし，その入力しているキーが変更される頻度は人間プレイヤの方があきらかに少ないという印象を観察から得た．よって入力キー種類の切り替え頻度を減らすことでより人間らしさに接近できる可能性があると考えている．

6.5 結論・今後の予定

本研究ではシューティングの人間らしいプレイヤ実現のための手法提案を行った．アクションや FPS の既存研究で扱われた諸手法をシューティングへの適用する接近法も考えられるが，ゲームの性質の相違（明確な目的地の不在によるキャラクタ動作の自由度の大きさ）から，簡単には成功しないと本研究では考えた．そのためシューティングの既存ゲームプレイヤの観察を通じて接近法を考案し，経路探索をベースにして Influence Map による被弾危険度見積もりと探索の諸工夫による手法を設計した．また実装と実験によって，Influence Map や探索の諸工夫が人間らしい印象に貢献することを確かめた．

とはいえまだ実際の人間には一段劣る性能だったので，改善の必要がある．特に人工プレイヤの動きのせわしなさを理由に人間プレイヤを高く評価するケースが目立ったため，移動の切り替えの頻度をなるべく抑えるべく，入力キーの切り替えに適度なペナルティを課すアプローチ等を試みるべきと考える．

また，シューティングの回避手段は移動だけでなく，ボムの使用によっても可能である．さらにプレイヤは常に回避に専念すれば良いわけではなく，敵の殲滅とのバランスも考えなくてはいけない．よって，ショットやボムの使用可能性を想定した条件に設計を拡大していくのも重要な発展課題である．

6.6 付録：使用したプレイヤの設計や実験条件の詳細

稿の構成としてはやや変則的であるが，本章で使用した手法に関するより細かな設計やパラメータなどをここで説明する．

6.6.1 経路の評価式

経路の評価値は各ノードの危険度を重ね付けして合計される．経路の評価値は経路上で t 番目に通るノード n_t の評価値を $E(n_t)$ として以下の式で表される．

$$\sum_{k=1}^{T} w^{(T-k)} E(n_t)$$

ただし 1 つの経路に含まれるノードの数を T とし，w は 1 未満の定数である．本稿では $w = 0.003$ とした．この w の決定指針は，$E(n_t)$ の値域がおよそ $[-350, 0]$ なので，t の小さいノードでの評価値がタイブレーク時の決定にのみなるべく用いられるように決めた．

6.6.2 Influence Mapの式

ゲーム画面座標 (x_g, y_g) を代表点とするInfluence Mapのグリッドが，画面座標 (x_b, y_b) にあって速度ベクトル (v_x, v_y) で運動する，当り判定半径（キャラクタの中心点座標との距離がこの値以下なら被弾）が R_{hit} である弾か敵から受ける被弾危険度の値について記す．計算時間節約のためルート演算を極力避けていることに注意されたい．

まず円形に分布する成分 D_{ci} を，

$$D_{ci} = I_{max} * max(0, 1 - \frac{(x_g - x_b)^2 + (y_g - y_b)^2}{R_{hit}^2 * C_{ci}^2})$$

と計算する．ただし I_{max} と C_{ci} は定数で，本稿ではそれぞれ50と3である．$max(a, b)$ は a と b のうちの大きい数値を表す記号である．

次に三角形に分布する成分 D_{tr} の計算のためには，まず (x_g, y_g) 弾からの相対位置座標 $(x_g - x_b, y_g - y_b)$ を，速度ベクトル (v_x, v_y) に沿う成分 m と直交する成分 s で表す．この m, n を以下のように定める．

$$\begin{pmatrix} m \\ s \end{pmatrix} = \begin{pmatrix} v_x & -v_y \\ v_y & v_x \end{pmatrix}^{-1} \begin{pmatrix} x_g - x_b \\ y_g - y_b \end{pmatrix}$$

ただし $v_x = 0$ かつ $v_y = 0$ の場合は $D_{tr} = 0$ とする．また $m < 0$ の場合も，$D_{tr} = 0$ と定める．その他の場合は，

$$D_{tr} = I_{max} * max(0, 1 - \frac{m}{C_{tr1}}) * max(0, 1 - \frac{|s|}{m * C_{tr2}})$$

とする．C_{tr1} は，速度方向への減衰を支配する定数で，本稿では90，C_{tr2} は三角形の角度を決める定数で本稿では $\tan 20^\circ$ とした．これらによって被弾危険度を $max(D_{ci}, D_{tr})$ とした．

また記述の煩雑を避けるため本稿では書かなかった工夫だが，画面の上側と左右の端近くには微細な被弾危険度を常に加算している．多くのシューティングでそれらの場所では敵が突然現れうるので，それをAIが避けるための措置である．

6.6.3 Influence Mapの補間

キャラクタ位置 (x_c, y_c) に割り当てる被弾危険度について述べる．(x_c, y_c) の最寄りのグリッド中心点を (x_{g0}, y_{g0}) として，x 座標が x_{g0} の点の中で (x_c, y_c) にその次に近いグリッド中心点を (x_{g0}, y_{g1})，y 座標が y_{g0} の点で (x_c, y_c) にその次に近いグリッド中心点を (x_{g1}, y_{g0}) とする．

もし $(x_c - x_{g0}, y_c - y_{g0})$ の，x軸との成す角とy軸との成す角の大きさが共に一定値 C_{an}(本稿では $\arctan(\frac{1}{2})$ とした) を超える場合に，(x_c, y_c) の被弾危険度は，

$$I(x_{g0}, y_{g0}) + (I(x_{g1}, y_{g1}) - I(x_{g0}, y_{g0})) * \frac{|x_c - x_{g0}| + |y_c - y_{g0}|}{|x_{g1} - x_{g0}| + |y_{g1} - y_{g0}|}$$

とする．ただし $I(x_g, y_g)$ はグリッド中心点 (x_g, y_g) に割り当てられた被弾危険度である．
それ以外の場合は，

$$d_x = (I(x_{g1}, y_{g0}) - I(x_{g0}, y_{g0})) * \frac{|x_c - x_{g0}|}{|x_{g1} - x_{g0}|}$$

$$d_y = (I(x_{g0}, y_{g1}) - I(x_{g0}, y_{g0})) * \frac{|y_c - y_{g0}|}{|y_{g1} - y_{g0}|}$$

として $I(x_{g0}, y_{g0} + d_x + d_y)$ とした．

6.6.4 ノード評価値の詳細

各ノードの詳細な評価値の式について示す．そのノードのキャラクタ位置 (x_c, y_c) に対する 6.6.3 の被弾危険度 $I(x_c, y_c)$ と，弾の近くを通った回数を用いて計算される．この『弾の近くを通った回数』の利用は説明の簡便さのため本文では記述を省いた．

『弾の近くを通った回数』とは，親ノード状態から移動してくる過程中，弾か敵に当たり判定の 2 倍の距離以内に近づいてしまった回数のことである．これは，全ての弾と敵，移動中の毎フレーム（複数フレームの強制移動時も）について合計する．この回数を $N_{nearlyHit}$ とし，ノードの評価値を

$$-I(x_c, y_c) - 50 * min(6, N_{nearlyHit})$$

と本稿では定めた．ただし $min(a, b)$ は a, b のうち小さい方の数値を表す．

6.6.5 実験 AI プレイヤ等のパラメータ設定

本研究が実験で使用したプレイヤ，Base 探索，I-Map，I-Map+複数 F のパラメータについて述べる．まず，各人工プレイヤが共通して備える「反射神経を模した障害物制御」だが，どの人工プレイヤでも 0.4 秒以内に生成された弾や敵を各人工プレイヤは探索時に考慮しない．

Base 探索は深さ 5 までの経路探索を行う．ただし，被弾するノード以外はノード評価値が常に 0 である．よって被弾の危険が全くない場合にとる移動行動は実装に左右されるが，今回の実装ではその場合「無移動」を出力する．

I-Map は深さに関して同様の数値設定であるが，6.6.2 節から 6.6.4 節に述べたような方法でノード評価に Influence Map を用いる．I-Map+複数 F は 5 フレームにまたがる移動を深さ 1 つ分として，深さ 5 まで探索を行う．ノード評価に Influence Map を用いる．

第7章　まとめ

本研究では未だ達成されていない対象ゲームと人工プレイヤの目的の組み合わせについて，いくつかの問題クラスが含む課題に対処するための拡張手法を提案し，性能を実験で評価した．ターン制ストラテジーでは強さを目的とし，既存のモンテカルロ型のアプローチに対し前向き枝刈りつきの $\alpha\beta$ 型の木探索を提案し，大会優勝プログラムに有意に勝ち越すプレイヤの開発に成功した．RPG では人間の好みの読み取りと迎合を狙う手法を新規に提案し，アンケートで不満の減少を確かめた．格闘ゲームでは強いプレイヤ作成を目指して既存の人工プレイヤ手法の長所と短所に注目し，複数の人工プレイヤを組み合わせるシステムの提案を行った．元となる人工プレイヤと比べての強さの向上が確かめられ，また「行動パターンの読み取られにくさ」が本手法によって人工プレイヤに付与されると予想される．シューティングでは人間らしいプレイヤ作成のため，アクションや FPS の既存手法の流用でなく，シューティングの既存プレイヤの挙動観察から手法の設計を図った．アンケートにより提案手法がシューティングにおいて人間らしさに寄与することを確かめた．

本研究のリサーチクエスチョンを以下に改めて示す．

1. 組合せによる合法手数が爆発的に増えるゲームで強い木探索プレイヤをどのように開発するのか
2. 人間プレイヤの個人ごとに異なるゲームスタイルの嗜好をどう読み取って迎合すれば良いか
3. 最善戦略が相手の行動によって変わり続けるリアルタイムゲームで人工プレイヤの強さを向上させるにはどうすれば良いのか
4. 人工プレイヤがシューティングゲームの障害物を回避する動きを人間らしくするにはどうすれば良いか

この最初の問いに関しては，本研究では一部のターン制ストラテジーゲームにおける諸性質に着目することで枝刈りを行い，TUBSTAP プラットフォーム上で既存のものより強い人工プレイヤの開発に成功した．その結果として以下の答えをこの問いに対し立てる．ターン制ストラテジーまたは手番ごとの複数の駒操作による組合せで合法手数が増えるターン制ゲームにおいて，「駒操作の順序に対する前後可能性」，「似た効果が期待される行動群の存在」，「少ない駒による複数回の木探索に探索を分割できそうな局面の多さ」の

うち複数個の性質を持ち合わせているゲームについては，本研究の枝刈り手法によって木探索型人工プレイヤの強さ向上を達成できる見込みがあると本研究では考える．

次の問に関しては，効用関数による人間嗜好のモデル化とシミュレーションによって，RPG ゲームで仲間の意図に沿った行動をとる人工プレイヤの開発を行った．その性能を自作の RPG プラットフォームの被験者実験により確かめた．そのため，この問いについては以下の答えを立てる．ゲームの状況と行動に関する人間の嗜好が関数でモデル化できて，人間の行動選択とその結果のシミュレーションがその嗜好の内容を特定するほどの情報量を含む場合ならば，本研究の提案手法によって個人の嗜好に迎合する仲間プレイヤの作成が可能であると考えられる．

３つ目の問いに対し，本研究では格闘ゲームを対象として既存のルールベースド型プレイヤを切り替えて使用するシステムの提案を行った．このシステムは，時刻ごとのキャラクタ体力の増減量に着目し，現在の敵に対して有望そうなプレイヤを SW-UCB アルゴリズムによって見出し，キャラクタの操作を任せる．FightingICE プラットフォームを用いた実験で提案手法が人工プレイヤの強さ向上に貢献することを確かめた．よって以下の答えをこの問いに対して立てる．格闘ゲームなど，最善戦略が相手の行動により変わり，なおかつある程度の複雑な連続的な行動系列が高い効果を持ち，さらに利用可能な計算時間に制約が強いようなリアルタイム制ゲームにおいて,「もしも既存の強いルールベース型人工プレイヤが複数用意できるのならば」，それらを切り替えることで元より強い人工プレイヤが作成できる見込みがある．

最後の問いに関してはシューティンゲームの自作環境で influence map とキー切り替え頻度の制限を用いた木探索によるプレイヤ作成を行い，その性能を被験者実験で評価した．そのため，この問いに対して本研究では以下の答えを立てる．沢山の障害物の回避の動きに細かさと高い自由度があるリアルタイム制のアクション型ゲームにおいて，その回避にのみ目的を絞った状況ならば，提案手法による経路探索によってある程度の人間らしさを備えた移動動作を人工プレイヤが行える見込みがある．

各領域での提案手法を比較したとき，他の領域にまで横断して応用できそうな知見は得られなかった．しかし各手法はその領域の中では他のゲームジャンルにも応用できる可能性がある．そのため本研究では４つの領域に属するこれらの問いが，後々にそれぞれの領域内に含まれるほぼ全ての課題の解決へと貢献していくことを期待する．本研究が現在取り扱ってきた課題は未解決なもののうちのほんの一部であるが，現存するゲームはだいたい大まかなジャンルで整理されており，各ゲームの性質もジャンル内である程度共通するため，そのような課題にはある程度の限りがあると本研究では考えている．今後そうした課題のうち目立つものから何かしらの対処を試みていくことが本研究の Future Work であると考えている．

関連図書

[1] Campbell Murray A. Joseph Hoane Jr, and Feng-hsiung Hsu. Deep blue. Artificial intelligence, 134.1 pp.57-83, 2002.

[2] 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. 第 11 回ゲームプログラミングワークショップ, pp.78-83, 2006.

[3] David Silver, et al. Mastering the game of Go with deep neural networks and tree search. Nature, 529.7587 pp.484-489, 2016.

[4] Hai Nan, et al. Turn-Based War Chess Model and Its Search Algorithm per Turn. International Journal of Computer Games Technology, 2016.5216861, 2016.

[5] Feiyu Lu, et al. Fighting Game Artificial Intelligence Competition Platform, IEEE 2nd Global Conference on Consumer Electronics, pp. 320-323, 2013.

[6] StarCraft II Official Game Site, https://starcraft2.com/ (2018/01/04).

[7] Claude E Shannon. Programming a Computer for Playing Chess, Philosophical Magazine Ser.7 41-314, 1950

[8] Arthur L Samuel. Some studies in machine learning using the game of checkers, IBM Journal of research and development44.1.2 (2000): 206-226.

[9] Jonathan Schaeffer, et al. Checkers is solved, science 317(5844), pp.1518-1522, 2007.

[10] R.U. Gasser. Solving Nine Men's Morris, Games of No Chance, pp.101-113, 1996.

[11] L.V. Ailis, et al. Go-Moku Solved by New Search Techniques. Computational Intelligence 12(1), pp.7-23, 1996.

[12] Michael Buro. Logistello: A strong learning othello program. Annual Conference Gesellschaft fur Klassifikation eV. pp.1-3, 1995.

[13] コンピュータ将棋プロジェクトの終了宣言, http://www.ipsj.or.jp/50anv/shogi/20151011.html/ (2018/02/03).

[14] Michael Genesereth, et al. General game playing: Overview of the AAAI competition.” AI magazine 26(2), pp.62-72, 2005.

[15] Gerald Tesauro. Td-gammon: A self-teaching backgammon program. Applications of Neural Networks, pp.267-285, 1995.

[16] 水上直紀, 他. 期待最終順位に基づくコンピュータ麻雀プレイヤの構築. 第 20 回ゲームプログラミングワークショップ pp.179-186, 2015.

[17] 池田心, 橋本隼一, and 土井佑紀. モンテカルロ + UCT における探索木のだまし構造. 第 24 回ゲーム情報学研究会, pp.1-4, 2010.

[18] IEEE CIG StarCraft AI Competition, https://cilab.sejong.ac.kr/sc_competition/ (2018/01/04).

[19] The GVG-AI Competition, http://gvgai.net/ (2018/01/04).

[20] 松原仁, 他. ロボカップ ロボカップの歴史と 2002 年への展望. 日本ロボット学会誌 20(1), pp.2-6, 2002.

[21] ai4wdcar, https://sites.google.com/site/ai4wdcar/home/ (2018/02/03).

[22] 鳥海不二夫, et al. 人狼知能サーバの構築. 第 19 回ゲームプログラミングワークショップ pp. 127-132, 2014.

[23] Kokolo Ikeda, et al. Production of various strategies and position control for Monte-Carlo Go Entertaining human players. Computational Intelligence in Games, pp.1-8, 2013.

[24] Julian Togelius, et al. Assessing believability. Believable bots, pp.215-230, 2013.

[25] Kokolo Ikeda, et al. Detection and labeling of bad moves for coaching go. Computational Intelligence and Games, pp.1-8, 2016.

[26] Miwa, Kazuhisa, et al. Tradeoff between problem-solving and learning goals: Two experiments for demonstrating assistance dilemma. Proceedings of the Cognitive Science Society 34(34), pp.2008-2013, 2012.

[27] Manuel Kerssemakers, et al. A procedural procedural level generator generator. Computational Intelligence and Games, pp. 335-341, 2012.

[28] Christopher Chang, et al. A Difficulty Metric and Puzzle Generator for Sudoku, UMAPJournal, pp.305-326, 2007.

[29] Sutiono, Arie Pratama, Ayu Purwarianti, and Hiroyuki Iida. A mathematical model of game refinement, International Conference on Intelligent Technologies for Interactive Entertainment, pp.148-151, 2014.

[30] Martin Zinkevich, et al. Regret minimization in games with incomplete information. Advances in neural information processing systems, pp.1729-1736, 2008.

[31] 五十嵐治一, 他. 方策勾配法による静的局面評価関数の強化学習についての一考察. 第 17 回ゲームプログラミングワークショップ, pp.118-121, 2012.

[32] Gerald Tesauro. Neurogammon wins computer olympiad, Neural Computation 1(3), pp.321-323, 1989.

[33] Guillaume Chaslot, et al. Monte-Carlo Tree Search: A New Framework for Game AI, Artificial Intelligence an Interactive Digital Entertainment Conference. pp.216-217, 2008.

[34] Volodymyr Mnih, et al. Playing Atari with Deep Reinforcement Learning, NIPS Deep Learning Workshop, 2013.

[35] Naoyuki Sato and Kokolo Ikeda. Three Types of Forward Pruning Techniques to Apply Alpha Beta Algorithm to Turn-Based Strategy Game, IEEE Conference on Computational Intelligence and Games, pp.294-301, 2016.

[36] Naoyuki Sato, et al. Estimation of Player's Preference for Cooperative RPGs Using Multi-Strategy Monte-Carlo Method, IEEE Conference on Computational Intelligence and Games, pp.51-59, 2015.

[37] 和田　堯之，他. 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, 第 33 回ゲーム情報学 (GI) 研究報告 pp.1-8, 2015.

[38] 和田　堯之. 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修士論文, 2015.

[39] Naoyuki Sato, et al. Adaptive Fighting Game Computer Player by Switching Multiple Rule-based Controllers, 3rd International Conference on Applied Computing and Information Technology, 2015.

[40] 佐藤　直之, 他．Influence Map を用いた経路探索による人間らしい弾避けのシューティングゲーム AI プレイヤ, 第 21 回ゲームプログラミングワークショップ, pp.57-64, 2016-09.

[41] Kunihito Hoki and Kaneko Tomoyuki. Large-Scale Optiization for Evaluation Functions with Minimax Search, Artificial Intelligence Research, 49, pp.527-568, 2014.

[42] Richard D. Greenblatt, et al. The Greenblatt Chess Program, Fall Joint Computing, pp.801-810, 1967.

[43] chessprogramming - Move Ordering, [Online]. Available: https://chessprogramming.wikispaces.com/Move+Ordering, [Accessed: 2016-04-30].

[44] Jonathan Schaeffer. Experiments in search and knowledge, Ph.D. Thesis, University of Waterloo, 1986.

[45] G. M. Adelson-Velskie, et al. Some methods of controlling the tree search in chess programs, Artificial Intelligence, 6.4, pp.361-371, 1975.

[46] Tord Romstat. An introduction to late move reductions. [Online]. Available: http://www.glaurungchess.com/lmr.html, [Accessed: 30- Apr- 2016].

[47] Kunihito Hoki and Masakazu Muramatsu. Efficiency of three forward-pruning techniques in shogi: Futility pruning, null-move pruning, and Late Move Reduction (LMR), Entertainment Computing, 3.3, pp.51-57, 2012.

[48] Alexander Reinefeld. An improvement of the Scout tree-search algorithm, ICCA Journal, 6.4, pp.4-14, 1983.

[49] Michael Buro. Probcut: An effective selective extension of the alphabeta algorithm, Icca Journal 18(2), pp.71-76, 1995.

[50] 鶴岡慶雅, 他. 局面の実現確率に基づくゲーム木探索アルゴリズム. 第 6 回ゲームプログラミングワークショップ, pp.17-24, 2001.

[51] Lieberum, Jens. An evaluation function for the game of amazons. Theoretical computer science 349.2, pp.230-244, 2005.

[52] Henry Avetisyan, and Richard J. Lorentz. Selective search in an Amazons program. Computers and Games 2002, pp.123-141, 2002.

[53] 川上裕生, 鶴岡慶雅. 多様な特徴量を用いた Arimaa 評価関数の比較学習, 第 19 回ゲームプログラミングワークショップ, pp.151-156, 2014.

[54] Julien Kloetzer, et al. The monte-carlo approach in amazons, Computer Games Workshop 2007, pp.185-192, 2007.

[55] David Jian Wu. Move Ranking and Evaluation in the game of Arimaa, Harvard University PhD thesis, 2011.

[56] Haizhi Zhong. Building a strong arimaa-playing program, University of Alberta PhD Thesis, 2005.

[57] David Fotland. Building a world-champion arimaa program, International Conference on Computers and Games, pp.175-186, 2004.

[58] Christ-Jan Cox. Analysis and implementation of the game arimaa, Universiteit Maastricht Master thesis, 2006.

[59] Sid Meier's Civilization Beyond Earth, [Online]. Available: https://www.civilization.com/jp/games/civilization-beyond-earth/. [Accessed: 19- Feb- 2016].

[60] Maurice Bergsma and Pieter Spronck. Adaptive Spatial Reasoning for Turn-based Strategy Games, AIIDE, pp.161-166, 2008.

[61] Stefan Wender and Ian Watson. Using reinforcement learning for city site selection in the turn-based strategy game Civilization IV, IEEE Symposium On Computational Intelligence and Games, pp.372-377, 2008.

[62] Christopher Amato and Guy Shani. High-level Reinforcement Learning in Strategy Games, Agent and Multi-Agent Systems, pp.75-82, 2010.

[63] Patrick Ulam, et al. Using Model-Based Reflection to Guide Reinforcement Learning, IJCAI Workshop on Reasoning, Representation, and Learning in Computer Games, pp.107-112, 2005.

[64] Tsubasa Fujiki, et al. A platform for Turn-Based Strategy Games, with a Comparison of Monte-Carlo Algorithms, IEEE Symposium On Computational Intelligence and Games, pp.407-414, 2015.

[65] 武藤 孝輔, 西野 順二. ターン制戦略ゲームにおけるファジィ評価を用いた探索木の枝刈り, 第 20 回ゲームプログラミングワークショップ, pp.54-60, 2015.

[66] TUBSTAP. [Online] (in Japanese). Available: http://www.jaist.ac.jp/is/labs/ikeda-lab/tbs_eng/index.htm [Accessed: 1- May- 2016]

[67] Nintendo, Advance Wars: Days of Ruin for Nintendo DS - Nintendo Game Details. [Online]. Available: http://www.nintendo.com/games/detail/nLeg9iJkPgq3fWBcqtpDNWUJ4IvmaQBY, [Accessed: 2- May- 2016].

[68] TUBSTAP Game AI Tournament 2016. [Online] (in Japanese). Available: http://www.jaist.ac.jp/is/labs/ikeda-lab/tbs/competition_menu.htm. [Accessed: 1- May- 2016]

[69] D. J. Edwards and T. P. Hart. The Alpha Beta Heuristic, MIT Artificial Intelligence Project Memo, 1963.

[70] Sander Bakkes, et al. TEAM : The Team-Oriented Evolutionary Adaptability Mechanism, Entertainment Computing, pp.273-282, 2004.

[71] Nobuto Fujii, et al. Evaluating Human-like Behaviors of Video-Game Agents Autonomously Acquired with Biological Constraints, Advances in Computer Entertainment Technology, pp.61-76, 2013.

[72] Mario AI Championchip 2012. [Online]. Available: http://www.marioai.org/

[73] ベルナッキャ マッテオ，星野 准一．AI platform for supporting believable combat in role-playing games，第 19 回 ゲームプログラミングワークショップ, pp.139-144, 2014.

[74] ベルナッキャ マッテオ，星野 准一．Believable fighting characters in role-playing games using the BDI model，情報処理学会第 31 回全国大会, pp.1-8, 2015.

[75] Kokolo Ikeda and Viennot Simon. Efficiency of static knowledge bias in monte-carlo tree search, Computers and Games, pp.26-38, 2014.

[76] Johannes van der Wal. Stochastic dynamic programming, Ph. D. dissertation, Methematisch Centrum, 1980.

[77] Peter Jozef Jansen. Using knowledge about the opponent in game-tree search, Ph. D. thesis, Carnegie-Mellon University, 1992.

[78] David Carmel, Shaul Markovitch. Learning Models of Opponent's Strategy in Game Playing, Planning and Learning, pp.140-147, 1993.

[79] Hiroyuki Iida, et al. Opponent-Model search, Maastricht University Technical Report, CS-93-03, 1993.

[80] 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，第 11 回 ゲームプログラミングワークショップ, pp.78-83, 2006.

[81] Andrew Y. Ng, Stuart Russell. Algorithms for Inverse Reinforcement Learning, 17 th international conference Machine Learning, pp.663-670, 2000.

[82] John von Neumann, Oskar Morgenstern. Theory of games and econoic behavior, Princeton University Press, 1944.

[83] Yoshimasa Tsuruoka, et al. Game-Tree Search Algorithm Based On Realization Probability, International Computer Games Association Journal 25(3), pp.145-152, 2002.

[84] Satoshi Namai and Takeshi Ito. A Trial AI System with Its Suggestion of Kifuu (playing style) in Shogi, Technologies and Applications of Artificial Intelligence, pp.433-439, 2010.

[85] Remi Coulom. Computing Elo ratings of move patterns in the game of Go, International Computer Games Association Journal 30(4), pp.198-208, 2007.

[86] K. Yamamoto, S. Mizuno, C. Y. Chu and R. Thawonmas, "Deduction of Fighting-Game Countermeasures Using the k-Nearest Neighbor Algorithm and a Game Simulator", Computational Intelligence and Games (CIG), IEEE, pp.1-5, 2014.

[87] H. Park and K. Kim, "Learning to Play Fighting Game using Massive Play Data", Computational Intelligence and Games (CIG), IEEE, 2014.

[88] S. S. Saini, C. W. Dawson and P. W. H. Chung, "Mimicking player strategies in fighting games", Games Innovation Conference (IGIC), pp.44-47, 2011.

[89] S. E. Ortiz B. , K. Moriyama, K. Fukui, S. Kurihara and M. Numao, "Three-Subagent Adapting Architecture for Fighting Videogames", PRICAI 2010: Trends in Artificial Intelligence, Springer Berlin Heidelberg, pp.649-654, 2010.

[90] A. Garivier and E. Moulines, "On Upper-Confidence Bound Policies for Switching Bandit Problems", Algorithmic Learning Theory, Springer Berlin Heidelberg, 2011.

[91] R. Agrawal. "Sample mean based index policies with $O(logn)$ regret for the multi-armed bandit problem", Advances in Applied Probability, pp.1054-1078, 1995.

[92] T. Graepel, R. Herbrich and J. Gold, "Learning to fight", Proceedings of the International Conference on Computer Games: Artificial Intelligence, Design and Education, pp.193-200, 2004.

[93] S. Lueangrueangroj and V. Kotrajaras, "Real-Time Imitation Based Learning for Commercial Fighting Games", Proc. of Computer Games, Multimedia and Allied Technology 09, International Conference and Industry Symposium on Computer Games, Animation, Multimedia, IPTV, Edutainment and IT Security, 2009.

[94] J. Hoshino, A. Tanaka and K. Hamana, "The Fighting Game Character that Grows up by Imitation Learning", Transactions of Information Processing Society of Japan 49.7 (2008): pp.2539-2548, 2008.

[95] B. H. Cho, S. H. Jung, Y. R. Seong and H. R. Oh, “Exploiting Intelligence in Fighting Action Games Using Neural Networks”, IEICE transactions on information and systems 89.3 (2006): pp.1249-1256, 2006.

[96] Martin Zinkevich, et al. Regret Minimization in Games with Incomplete Information, Advances in neural information processing systems, pp.1729-1736, 2007.

[97] Mario AI championship Turing track, http://www.marioai. org/turing-test-track (2016/9/26).

[98] The 2k botprize, http://botprize.org/ (2016/9/26).

[99] 上海アリス幻樂団, http://www16.big.or.jp/ zun/ (2016/9/26).

[100] 龍神録プログラミングの館, http://dixq.net/rp/index.html (2016/9/26).

[101] 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 2010 pp.1-7, 2010.

[102] 池田心. モンテカルロ碁における多様な戦略の演出と形勢の制御: 接待碁 AI に向けて. ゲームプログラミングワークショップ論文集 2012 pp.47-54, 2012.

[103] Matteo Bernacchia and Hoshino Jun’ichi. Believable fighting characters in role-playing games using the BDI model, 第 33 回ゲーム情報学 (GI) 研究報告 pp.1-8, 2015.

[104] Mario Games, https://mario.nintendo.com/ (2016/9/26).

[105] Noor Shaker, et al. The Turing test track of the 2012 Mario AI championship: entries and evaluation, Computational Intelligence in Games (CIG 2013), pp.1-8, 2013.

[106] Nobuto Fujii and et al. Evaluating human-like behaviors of video-game agents autonomously acquired with biological constraints, Advances in Computer Entertainment pp.61-76, 2013.

[107] Daichi Hirono and Ruck Thawonmas. Implementation of a human-like bot in a first person shooter: second place bot at botprize 2008, Proc. on Asia Simulation Conference, 2009.

[108] Jacob Schrum, Karpov Igor V. and Miikkulainen Risto. UT2: Human-like behavior via neuroevolution of combat behavior and replay of human traces, Computational Intelligence and Games (CIG 2011), pp.329-226, 2011.

[109] Bernard Gorman and et al. Believability testing and bayesian imitation in interactive computer games, Simulation of Adaptive Behavior, pp.655-666, 2006.

[110] Bhuman Soni and Philip Hingston. Bots trained to play like a human are more fun, IEEE International Joint Conference on Neural Networks, pp.363-369, 2008.

[111] Wikipedia: グラディウスシリーズ, https://ja.wikipedia.org /wiki/グラディウスシリーズ (2016/9/26).

[112] Wikipedia: ゼビウス, https://ja.wikipedia.org/wiki/ゼビウス (2016/9/26).

[113] 桑谷拓哉, 橋本剛. 熟練プレイヤーレベルを目指す弾幕シューティング AI の開発. 情報科学技術フォーラム講演論文集 12.2 pp.383-384, 2013.

[114] 東方花映塚 Phantasmagoria of Flower View, http:// www16.big.or.jp/ zun/html/th09top.html (2016/9/26).

[115] Uriarte Alberto and Santiago Ontanon. Kiting in RTS games using influence maps, Eighth Artificial Intelligence and Interactive Digital Entertainment Conference. 2012.

[116] Yoram Koren and Johann Borenstein. Potential field methods and their inherent limitations for mobile robot navigation, Proceedings on Robotics and Automation, pp.1398-1404, 1991.

[117] Welcome to Fighting Game AI Competition, http://www. ice.ci.ritsumei.ac.jp/ ftgaic/ (2016/9/26).

[118] いで庵, http://www.usamimi.info/ ide/index.html (2016/9/26).

業績リスト

国内学会発表

1. 佐藤直之，池田心．花札のこいこいにおける方策勾配法と Neural Fitted Q Iteration の適用，第 22 回ゲームプログラミングワークショップ，査読あり，2017.

2. 佐藤 直之，池田 心，上原 隆平．花札の「こいこい」ゲームの強化学習によるコンピュータプレイヤ，情報処理学会 第 38 回ゲーム情報学研究，査読なし，2017，

3. 佐藤　直之, Sila Temsiririrkkul, Luong Huu Phuc, 池田　心．Influence Map を用いた経路探索による人間らしい弾避けのシューティングゲーム AI プレイヤ, 第 21 回ゲームプログラミングワークショップ，査読あり，2016.

4. 中川 絢太, 佐藤 直之, 池田 心．ゲームの目的達成のみを追求した AI では生まれにくいゲーム内行動の分類と考察，第 36 回ゲーム情報学研究会，査読なし，2016.

5. 萩原　涼太，山田　渉央，佐藤　直之，池田　心．麻雀における相手の和了点数予測法の性能評価，第 35 回ゲーム情報学研究会，査読なし，2016.

6. 佐藤直之，藤木翼，池田心．ターン制戦略ゲームにおける局面評価値構成のための局面分割および単純化ゲームのオフライン木探索，第 20 回ゲームプログラミングワークショップ，査読あり，2015.

7. 大町洋，佐藤直之，池田心．複数ソルバを用いた上海ゲームのインスタンス生成，第 18 回ゲームプログラミングワークショップ，査読あり (ポスター発表)，2013.

国際会議発表

1. Naoyuki Sato and Kokolo Ikeda. Three Types of Forward Pruning Techniques to Apply Alpha Beta Algorithm to Turn-Based Strategy Game, IEEE Conference on Computational Intelligence and Games, peer reviewed, 2016.

2. Kokolo Ikeda, Simon Viennot and Naoyuki Sato. Detection and Labeling of Bad Moves for Coaching Go, IEEE Conference on Computational Intelligence and Games, peer reviewed, 2016.

3. Naoyuki Sato, Kokolo Ikeda and Takayuki Wada. Estimation of Player's Preference for Cooperative RPGs Using Multi-Strategy Monte-Carlo Method, IEEE Conference on Computational Intelligence and Games, peer reviewed, 2015.

4. Naoyuki Sato, Sila Temsiririkkul. Shogo Sone and Kokolo Ikeda, Adaptive Fighting Game Computer Player by Switching Multiple Rule-based Controllers, 3rd International Conference on Applied Computing and Information Technology, peer reviewed, 2015.

5. Naoyuki Sato. Adapteive Fighting Game Computer Player by Switching Various Players, Biyani's International Conference in Jaipur, without peer review, 2015.

論文誌

1. 佐藤直之，藤木翼，池田心．戦術的ターン制ストラテジーゲームにおける AI 構成のための諸課題とそのアプローチ, 情報処理学会論文誌 57(11)，2016.