

2014年度 卒業論文

ターン制ストラテジーゲームにおける  
ファジー論理を用いた個性的なAI設計手法

指導教員：渡辺 大地 講師

三上 浩司 准教授

メディア学部 ゲームサイエンス プロジェクト

学籍番号 M0111341

布浦 拓実

2014年度 卒業論文概要

論文題目

ターン制ストラテジーゲームにおける  
ファジー論理を用いた個性的な AI 設計手法

メディア学部

学籍番号：M0111341

氏名

布浦 拓実

指導  
教員

渡辺 大地 講師  
三上 浩司 准教授

キーワード

ターン制ストラテジー, AI, 影響マップ, ファジー

ターン制ストラテジーゲームにおいて、プレイヤーに飽きを感じさせず、長くゲームを遊んでもらうために、個性的な思考パターンを持つコンピュータプレイヤー AI を複数用意することは重要である。AI が様々な戦術を駆使してくることで、プレイヤーも異なる戦い方をする必要が出るため、ゲームのワンパターン化を防げる。しかし、個性のある AI をいくつも設計することは時間がかかる。設計者が AI に個性を持たせるためには、AI の行動を定める様々な評価関数を作り、パラメータを調整したり評価関数同士を組み合わせる必要がある。これらの調整は直感的には行えず、試行錯誤を繰り返しながら理想の AI に近づけていくことになる。このように、1つの AI の設計に手間と労力がかかるため、個性的な AI をいくつも用意することは容易ではない。

本研究では、個性のある AI をより直感的に設計できるようにするために、影響マップとファジー論理を組み合わせた AI 設計手法について提案する。AI の行動の評価値を影響マップに入力する際、あいまいさを許容するファジー論理を用いることで、より設計者が直感的な AI 設計を行えるようにする。そして、影響マップのみの手法と提案手法で同じ AI を設計し比較することで、本手法の有用性を示す。

# 目次

第1章	はじめに	1
1.1	研究背景	1
1.2	論文構成	4
第2章	影響マップを用いた AI 設計手法について	5
第3章	提案手法	9
3.1	ファジー論理について	9
3.2	影響マップとファジー論理の組み合わせ	13
第4章	検証と考察	23
4.1	検証で用いるゲーム環境の解説	23
4.2	実験概要	24
4.3	実験結果	27
4.4	考察	30
第5章	まとめ	31
	謝辞	32
	参考文献	33

# 第 1 章

## はじめに

### 1.1 研究背景

コンピュータゲームのジャンルの1つとして、ターン制ストラテジーというものがある。これは、プレイヤー同士が互いに複数の駒を動かして戦い、目的の達成を目指すもので、シミュレーションゲームの一種である。既存のゲームとしては、ファミコンウォーズDS[1]、Civilization[2]などが該当する。ターン制ストラテジーにおいて、コンピュータプレイヤーのAIに個性を付けることは、プレイヤーに飽きや単調さを感じさせないために重要なことである。様々な異なる戦術を持つAIを対戦相手として用意することで、プレイヤーは全く同じ戦い方では勝つことができず、相手ごとに戦術を考え抜いた上で勝利を目指すという楽しみが生まれる。

AIの強さだけを調整するならば、候補手探索を最適化し、より良い手を選ばせるようにするか、あるいはあまり良くない手を選ばせるように調整することで行える。しかし上田ら [3] は、AIの強さだけを変えても、基準となる思考ルーチン自体は変わらないので、戦術や弱点は似通ってしまうと述べている。また、AIの行動選択に乱数を加えることでも、毎回違った行動をするAIを表現することができるが、それは一つの行動基準に沿って行動しているわけではない。「気まぐれなAI」という一つのAIの種類にはなるものの、個性的なAIを複数用意することには繋がらない。

本研究では上記の点を踏まえ、異なった価値観を持ち、他とは違う戦術を駆使する AI のことを「個性的な AI」と呼ぶ。個性的な AI をいくつも用意する場合、一つ一つの AI の考え方に個性を付けていくことになる。これを手作業で行うのには手間がかかるので、自動化できれば理想的である。その手法の一つとして、機械学習という手法がある。機械学習とは、設計者が全ての動作をプログラムせずとも、コンピュータにデータを与えることでコンピュータ自身が法則性などを見つけ出し、行動を学習するという人工知能の研究分野の一つである。

チェスや囲碁、将棋などの古典的なボードゲームの分野では、AI の個性や棋風を機械学習で実現する研究を多くの研究者が行っている。生井ら [4] は、プロ棋士が持つ棋風を分析して評価関数に組み込む研究を行い、序盤の定石選択確率を模倣した AI を実現した。仲道ら [5] は、将棋の機械学習にアマチュアレベルの棋譜データを用いることで、人間らしい適度な弱さを持つ AI の実現を目指す研究を行った。志水ら [6] は、UCT に *prior* を用いることで、どうぶつしょうぎにおいて個性的な AI を実現した。他にも、機械学習により AI に個性を持たせる研究は数多く存在する。これらは、長年の将棋の研究により積み重なった定石や、プロ棋士達による膨大な数の棋譜など、様々なデータを教師データとして用いている。

古典的なボードゲームの分野と比較すると、ターン制ストラテジーにおける AI の研究の数は少ない。加藤ら [7] は UCT 探索において無駄な移動行動を枝刈りすることで探索を効率化し、単純な UCT よりも強い AI を実現している。藤木ら [8] は、攻撃的な行動の探索が得意な攻撃行動探索、陣形を重視する防御的な DLMC、そしてそれらを組み合わせた UCT での探索手法を提案し、従来よりも強い AI を実現している。このように、ターン制ストラテジーにおける強い AI の実現を目指す研究はあるものの、機械学習などを用いて個性的な AI を自動調整するような研究は進んでいない。

ターン制ストラテジーにおける AI 研究があまり進まない理由の 1 つとして、統一された環境での研究が難しいという理由がある。同じターン制ストラテジーというジャンルの中でも、個別のゲームごとにルールが異なるため、どのような手

を指すことが有利・不利に繋がるのかも、ゲームごとに異なってしまふ。そのため、AIの設計手法も千差万別になってしまい、統一されたルールを持つチェスなどよりも研究がし辛いという現状がある。三宅 [9] は、シミュレーションゲームなどに応用される地形解析技術が、現状では特定のゲームに合わせたニッチな手法となっていると述べている。

また、チェスなどでの研究では定石や棋譜などの膨大なデータを利用することができるが、ターン制ストラテジーでは、開発中のゲームは全く新しいゲームということになるので、発売前に大量の棋譜を用意することは困難である。また、ターン制ストラテジーは元々コンピュータプレイヤーと戦うことが前提のジャンルであるため、AIに求める考え方の個性と人間が持つ考え方の個性が必ずしも一致しないという問題もある。

これらの理由から、現状ではターン制ストラテジーにおける個性的なAIの設計を機械学習で行うことは難しく、ゲームの設計者が手作業で行うことになる。設計者は、思い描く挙動をするAIを実現するため、様々な評価関数を作ったり、組み合わせたり、パラメータを細かく調整したりする。これらの調整は直感的に行えるものではなく、何度も試行錯誤をする必要があり、大きな手間と労力がかかる。

本研究では、ターン制ストラテジーにおける個性的なAIを設計する際、より直感的な設計を可能にし手間と労力を減らすことを目的として、影響マップとファジー論理を組み合わせたAI設計手法を提案する。影響マップとは、ゲームのマップにおいて、ある地点が周りの環境からどれくらいの影響を受けているかの情報を表すものである。ファジー論理とは、真と偽に加えその間のあいまいな数値も扱える、多値論理の1つである。

影響マップはターン制ストラテジーのAI設計に広く使われており、個性のあるAIを実現するためにも有用な手法であるが、これだけでは直感的なAI設計が行えず、パラメータ調整に試行錯誤が必要となる。本研究では、あいまいな入力から数値の出力ができるファジー論理を影響マップの評価値入力に用いる。これにより、設計者は想定通りの影響マップをより直感的に生成できるようになり、個

性的な AI の設計にかかる手間を削減することができる。そして、影響マップのみを用いた既存手法と比較検証をすることで、本手法の有用性を示す。

## 1.2 論文構成

本論文は、本章を含めて全 5 章で構成する。第 2 章では、影響マップのみを用いた既存の AI 設計手法と、その問題点について述べる。第 3 章で提案手法について述べ、第 4 章では提案手法と既存手法の比較検証を行う。第 5 章では本研究のまとめを行う。

## 第 2 章

# 影響マップを用いた AI 設計手法について

本章では、影響マップを単体で用いる従来の AI 設計手法とその問題点について述べる。ターン制ストラテジーの AI は、現在の状況に応じて、自分が指せる手の一つ一つがどれくらい良い手であるかを判断し、最終的に指す手を決める。こういった、状況を受け取り自身の行動を返す処理を行うものを評価関数 [10] と呼ぶ。評価関数を使ってそれぞれの手の良さを調べるためには、AI が盤面の状態を把握することが必要となる。敵・味方の位置や数、種類などを知ることによって、「ここは敵が多いので不利である」といったような判断ができるのである。

盤面の状態を表す方法として、影響マップ [11] という手法がある。影響マップとは、ゲームのマップ上において、ある地点が周囲の環境からどの程度影響を受けているのかを表したものである。影響マップは、リアルタイムストラテジーの経路探索 [12] で用いることが多い手法であるが、マップ上のマス有利度などを表せるため、ターン制ストラテジーでの意思決定 [13] にも使うことができる。次の図 2.1 は、影響マップの例である。



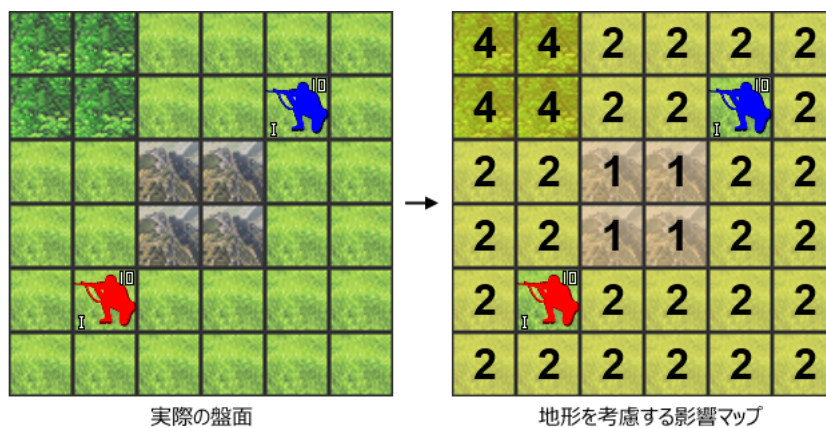


図 2.1: 影響マップの例

マスに書かれている数値は、AIにとってその位置へ移動することの評価の度合いを表しており、これを本論文では望ましさと呼ぶ。この例では、動きにくい山の望ましさを低いものとし、防御力が上がる森の望ましさを高いものとしている。AIは影響マップの情報をもとに、最も望ましさが高い行動を行う。また、生成した影響マップは、他の影響マップと組み合わせることができる。次の図 2.2 は、2つの影響マップを組み合わせ、1つの影響マップを生成した例である。

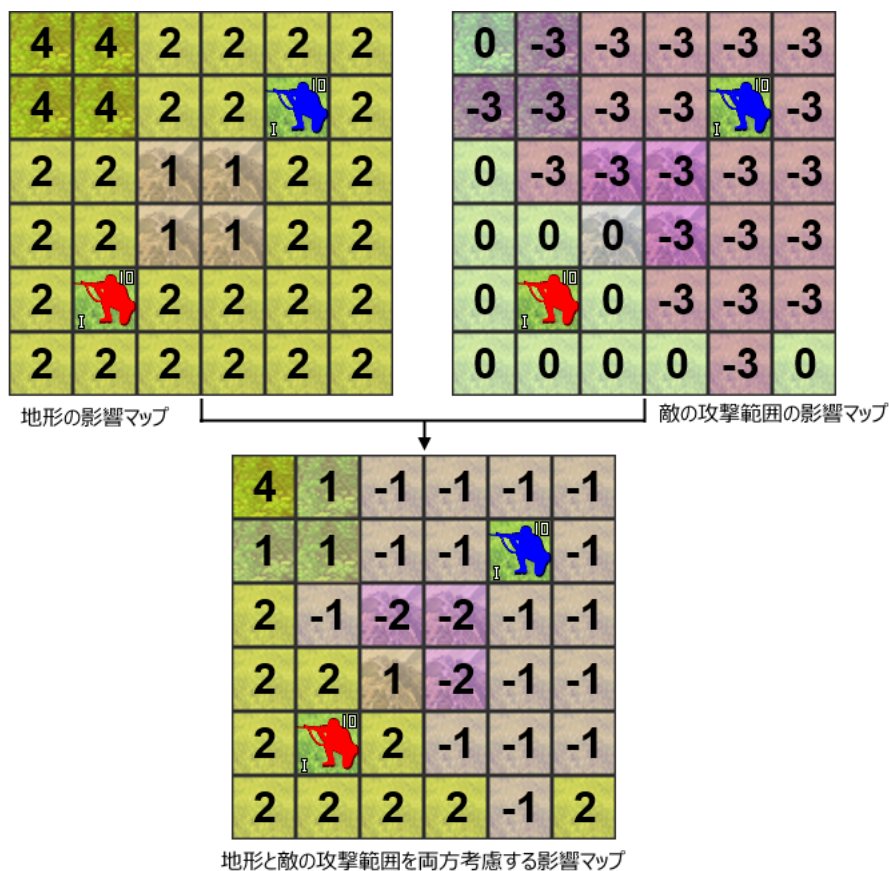


図 2.2: 2 つの影響マップを組み合わせた例

左上の影響マップは地形を考慮している。右上の影響マップは、赤いユニットを自分のユニットと想定した場合における、敵の攻撃範囲内にマイナスの評価をしている。この2つの影響マップを組み合わせ、同じマスの望ましさを単純に足し合わせることで、地形と敵の攻撃範囲を両方考慮した影響マップが生成できる。このように、AIに考慮させたい要素についての影響マップを用意し、足し合わせていくことで、様々なことを考慮して動くAIが作れる。AIに有利と考えさせたい部分には高い値を、不利と考えさせたい部分には低い値を入れることで、AIの好みや価値観を表現することができ、個性的なAI設計が行える。

しかし、影響マップに対し具体的にどんな値を入力すれば良いのかは直感的には分からず、様々な値を入力しながら試行錯誤することが必要となる。影響マップを組み合わせた例では、敵の攻撃範囲内のマスには悪い評価である「-3」を入力

しているが、この値が丁度良いということは試行錯誤して見つけ出さなければならぬ。また、この他に3つ、4つと新たな影響マップを組み合わせていくとすると、他の影響マップの値とうまくバランスを取らなければならぬので、より調整が難しくなっていき、大きな手間と労力がかかる。

そこで本研究では、影響マップに入力する値をファジー推論で計算することで、影響マップを自動的に生成する手法を提案する。これにより、設計者が影響マップに色々な値を入力する手間が省け、より直感的に個性的なAIの設計ができるようになる。提案手法の詳細は次章で述べる。

# 第 3 章

## 提案手法

本章では、ファジー論理について説明し、影響マップとファジー論理を組み合わせた提案手法について述べる。

### 3.1 ファジー論理について

ファジー論理 [14][15] とは、真と偽だけではなく、その間のあいまいな数値も扱える多値論理の 1 つである。ファジー論理では、人間が物事の度合いを判断するときの言語表現で、様々な分析や問題解決をすることができる。まず、従来のブール理論について述べる。ブール理論では、気温を「暑い」「丁度良い」「寒い」に分類したい時、何度以上は暑い、何度以下は寒いと、はっきりとした境界線を定義する必要がある。仮に 25 度以上を「暑い」とすると、25 度も 30 度も 40 度も等しく「暑い」という集合に属し、25 度未満は「暑い」には属さないこととなる。このように、ある数値が集合に属するか属さないか明確に分かれる集合を、クリスプ集合と呼ぶ。次の図 3.1 は、クリスプ集合の特徴関数である。

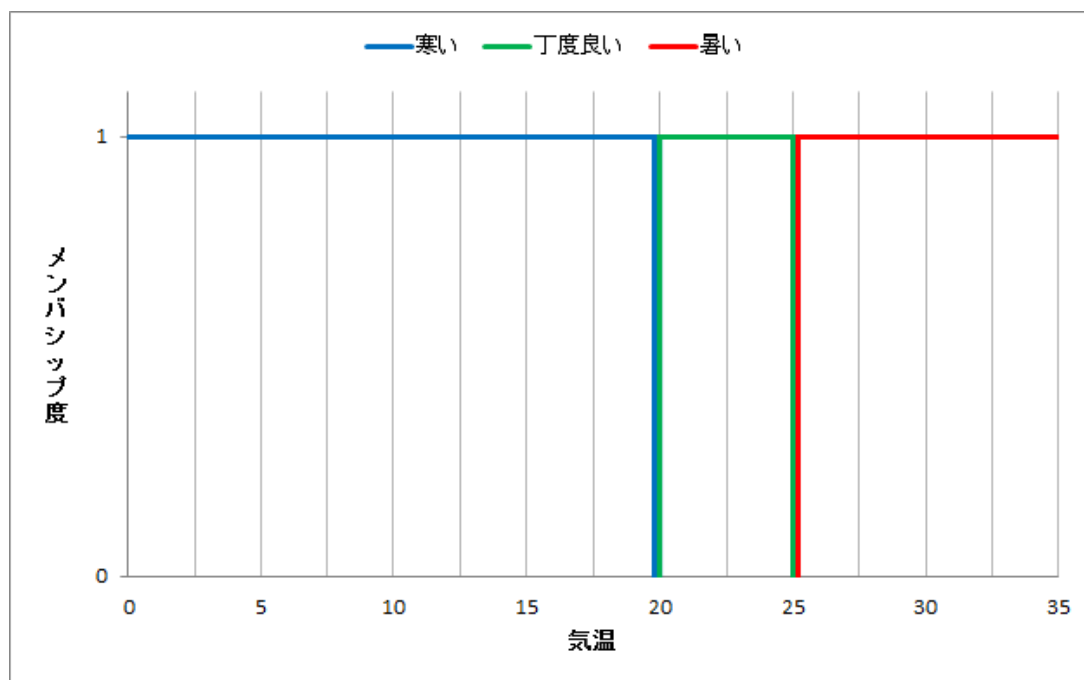


図 3.1: クリस्प集合の特徴関数

この特徴関数は、20 度未満を「寒い」、20 度以上 25 度以下を「丁度良い」、25 度を越えると「暑い」となることを表している。クリस्प集合であるため、どの気温においてもいずれかの集合に完全に属している。このクリस्प集合を用いて何かしらの制御をしようとした際、不都合な点が出てくることがある。25.1 度と 25 度は、0.1 度しか変わらないので体感的な差は小さいと推測できるが、25.1 度は「暑い」となり、25 度は「丁度良い」となる。また、15 度は 0 度と比べれば遥かに暖かく感じるはずだが、どちらも「寒い」という同等の評価しかできない。

このような、曖昧な度合いの問題を扱えるのがファジー論理である。ファジー論理では、集合の論理値に 0 以上 1 以下の実数値を扱うことができる。1 に近いほど真に近く、0 に近いほど偽に近いことを表す。こういった集合を、ファジー集合と呼ぶ。ファジー集合の特徴関数は、メンバーシップ関数と呼ぶ。次の図 3.2 は、気温をファジー集合で表したメンバーシップ関数である。

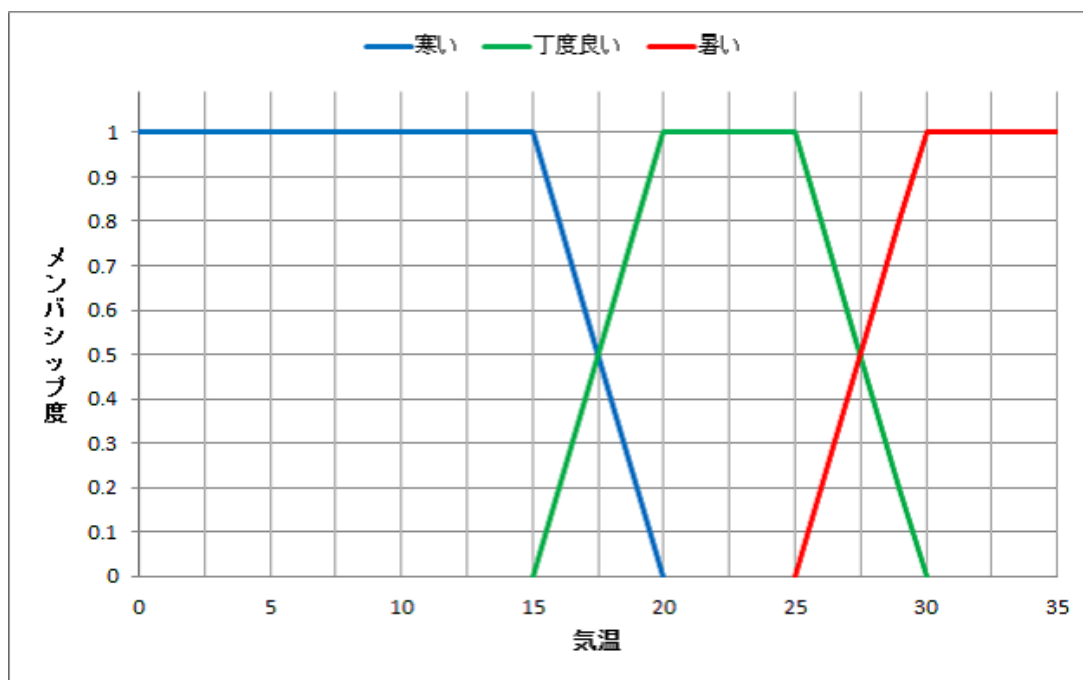


図 3.2: ファジー集合のメンバーシップ関数

この例では、15度～20度と25度～30度がそれぞれ2つの集合に属している。2つの集合に属している場合は、それぞれの集合に対して部分的に属していることを意味する。例えば27.5度は、「丁度良い」のメンバーシップ度が0.5、「暑い」のメンバーシップ度も0.5となっているため、丁度良いとも暑いとも取れるという評価になる。

曖昧な数値を扱えるファジー集合には様々な使用方法があり、その中の一つに、「ファジールールに基づく推論」がある(以下、ファジー推論)。これは、ファジー化された数値を用いて、曖昧な言語からなるファジールールから値を求め、クリスプ値を出力するというものである。ファジー推論のルールは、「もし～ならば、～する」という形のIF-THENルールで記述する。「もし～ならば」の部分を前件部と呼び、「～する」の部分を後件部と呼ぶ。ファジー推論では、前件部にも後件部にもファジー集合を用いることになる。例えば、室温と湿度を元に、クーラーの温度設定をする処理を想定する。条件となるのは室温と湿度なので、この2つが前件部となる。行動となるのはクーラーの温度設定なので、これが後件部とな

る。次の表 3.1 は、室温と湿度の組み合わせを元にルールを記述したものである。

表 3.1: 室温・湿度を条件としたファジールール

ルール	前件部	後件部
ルール 1	室温が高く、湿度も高い	設定温度を下げる
ルール 2	室温が高く、湿度が低い	設定温度は変えない
ルール 3	室温が低く、湿度が高い	設定温度は変えない
ルール 4	室温が低く、湿度も低い	設定温度を上げる

次に、室温・湿度・設定温度それぞれのファジー集合を用意する。この例では、室温・湿度が持つ集合はそれぞれ「高い」「丁度良い」「低い」の3つとし、設定温度が持つ集合は、「温度を下げる」「変えない」「温度を上げる」の3つとする。これで、ファジー集合とファジールールが用意できたので、室温と湿度の組み合わせを入力することで設定温度を出力することができる。室温・湿度共に「高い」のメンバシップ度が1.0の場合、ルール1に完全に適合しているので、設定温度を下げるという判断になる。このとき、設定温度を具体的に何度変えるかという度合いも導き出すことができる。室温・湿度共に「丁度良い」のメンバシップ度が0.7、「低い」のメンバシップ度が0.3の場合、ルール4に少しだけ適合しているので、肌寒いと判断して少しだけ設定温度を上げるという結果になる。

このように、ファジー推論を用いることで、現在の状況を曖昧なファジールールに照らし合わせて物事の判断を行い、意思決定をする処理が設計できる。ファジー論理は人工知能やロボット制御などに幅広く用いられており、ゲームAIの分野でも様々な研究が行われている。星野ら [16] は、チェスにおける強化学習にファジー環境評価ルールを用いることで、AIをより強くすることに成功した。大西 [17] は、ファジー論理を用いた直感的なゲームデザインの研究として、確率分布の生成によるゲームバランス調整の研究を行っている。

## 3.2 影響マップとファジー論理の組み合わせ

本研究では、影響マップに入力する評価値にファジー論理を用いることで、曖昧なルールから行動の評価値を出力し、直感的に影響マップを生成する手法を提案する。2章で述べた通り、影響マップだけでも個性的なAIの設計は可能であるが、それだけでは入力する評価値を直感的に設定できず、調整に手間と労力がかかる。そこで本手法では、AIが考慮する各評価項目について、どの程度を「良い」「普通」「悪い」と判断するのかをファジー集合で表し、ファジールールに従って最終的な評価値を出力する。例として、有利な際には攻撃をし、不利な際には逃げるAIを設計することを想定する。ゲームのルールは、互いにユニットを動かして、敵ユニットに攻撃してダメージを与え、全滅させれば勝利というものとする。この例で用いるファジー集合は、次の通りである。

- 攻撃の際、敵に与えるダメージ（前件部）
- 移動地点への、敵と味方の影響度（前件部）
- その行動の望ましさ（後件部）

「攻撃の際、敵に与えるダメージ」とは、敵ユニットの隣に移動した際、攻撃で与えるダメージの良し悪しを表す集合である。5ダメージ以上を「良い」、3ダメージを「普通」、1ダメージ以下を「悪い」に設定する。移動した際、敵ユニットが隣にいない場合は攻撃ができないが、その場合のメンバシップ度は「普通」が1.0となるものとする。与えるダメージのメンバシップ関数を図3.3に示す。



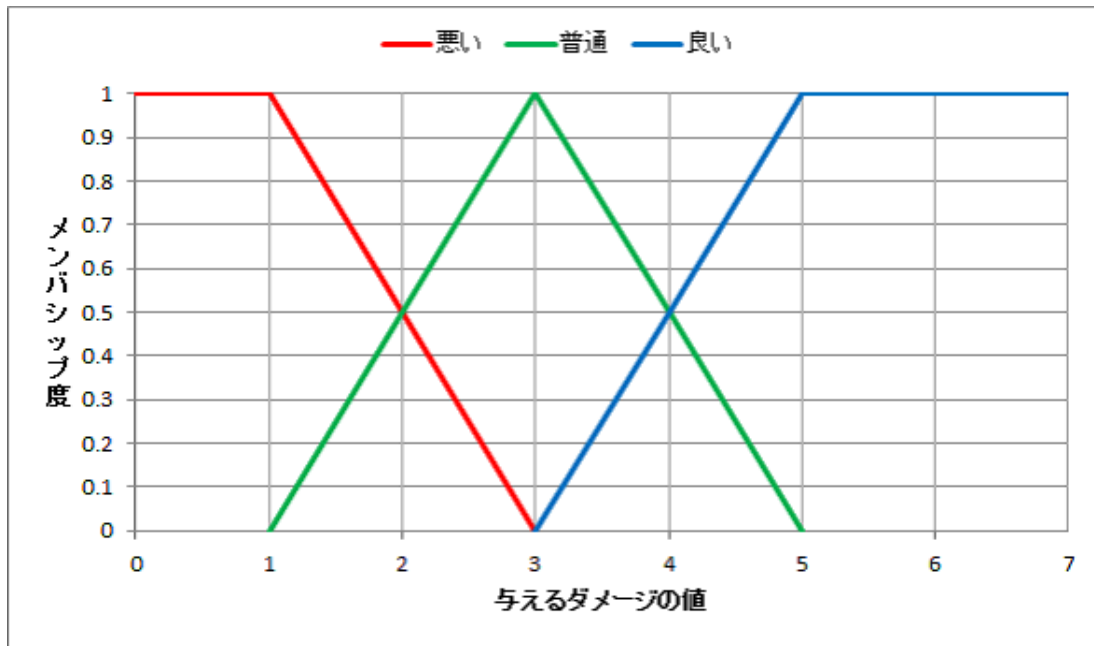


図 3.3: 与えるダメージのメンバシップ関数

「移動地点への、敵と味方の影響度」とは、移動先のマスに対して、味方と敵のユニットそれぞれがどの程度攻撃することができるかを表している。攻撃可能自軍ユニットの攻撃力の和から、攻撃可能敵軍ユニットの攻撃力の和を引いた数値を影響度とする。影響度が正の数値なら、味方の援護が多いことを意味し、影響度が負の数値なら、敵からの脅威が多いことを意味する。今回は、5以上を「良い」、0を「普通」、-5以下を「悪い」とする。その場合のメンバシップ関数を図 3.4 に示す。

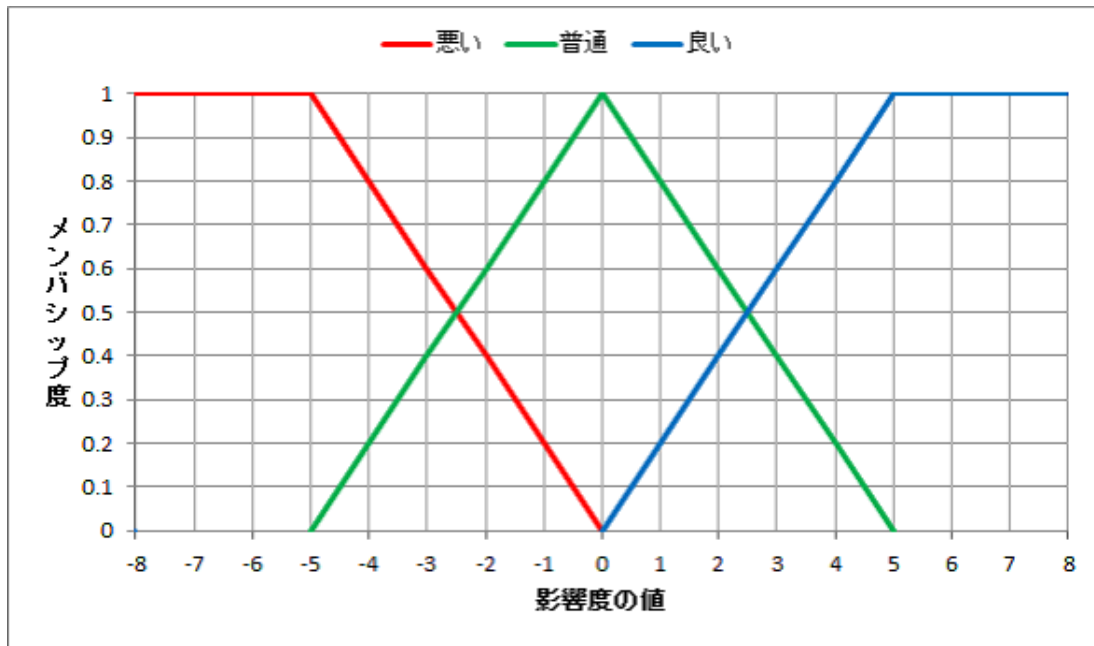


図 3.4: 敵と味方の影響度のメンバーシップ関数

最後に、後件部である「行動の望ましさ」のメンバーシップ関数を用意する。今回は、望ましさを 0 点 ~ 100 点で表すこととする。図 3.5 は、行動の望ましさを表すメンバーシップ関数である。

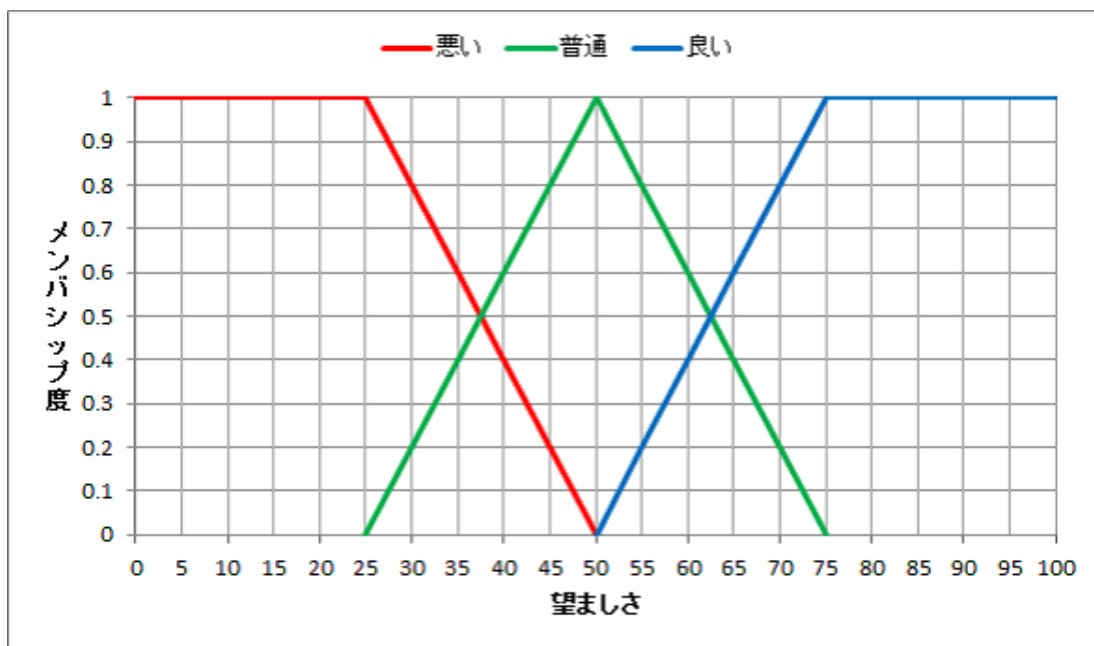


図 3.5: 望ましさのメンバーシップ関数

必要なファジー集合が用意できたので、次はファジールールを記述する。単純なファジー推論では、「与えるダメージが良くて、影響度も良いなら、望ましさは良い」「与えるダメージが良くて、影響度が普通なら、望ましさは普通」などというように、複数の前件部を条件として後件部の値を計算する。しかしこの方法では、集合や要素の数が少し増えるだけで、すぐに組み合わせ爆発を起こしてしまう。今回の例では、前件部が2つあり、それぞれ3つの集合を持っているので、9つのルールが必要になる。別の評価項目を追加しようと考えて前件部を1つ増やすと、必要なルールは27個に増える。更に前件部を増やしていくと、ルールの数は81個、243個と増加していく。高速な計算が要求されるゲームAIにおいて、組み合わせ爆発が起きてしまうことは致命的である。

そこで本手法では、William E.Combsが開発した「Combs手法」[18]を用いる。Combs手法では、1つのファジールールに1つの前件部のみを使う。「与えるダメージが良いなら、望ましさは良い」「影響度が悪いなら、望ましさは悪い」というように、前件部1つから後件部の値を出力する。この方法では、必要なファジールールの数は、(要素数) × (集合数)となるので、組み合わせ爆発が起こらなくなる。Combs手法に基づいて記述するファジールールは、次の表3.2の通りとする。

表 3.2: 与えるダメージ・影響度を条件としたファジールール

ルール	前件部	後件部
ルール1	与えるダメージが「良い」	望ましさは「良い」
ルール2	与えるダメージが「普通」	望ましさは「普通」
ルール3	与えるダメージが「悪い」	望ましさは「悪い」
ルール4	影響度が「良い」	望ましさは「良い」
ルール5	影響度が「普通」	望ましさは「普通」
ルール6	影響度が「悪い」	望ましさは「悪い」

ファジー集合とファジールールが用意できたので、ゲーム中の値から望ましさを出力することができる。実際のゲームの盤面を想定して、どのような計算を行うかについて述べる。例とする盤面の状態を図3.6で示す。

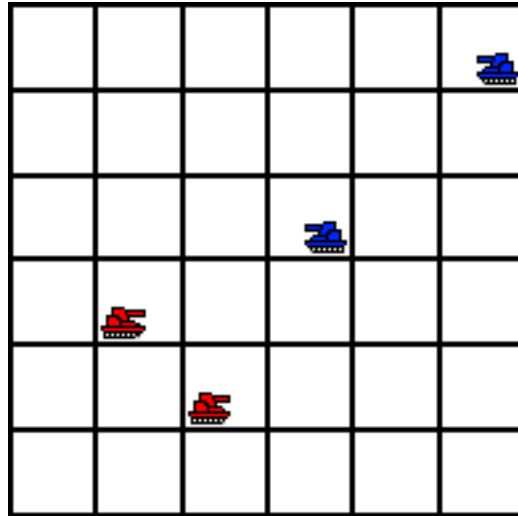


図 3.6: 盤面

盤面には正方形のマスが縦横それぞれ6マス敷き詰められている。一番左上のマスを(1,1)、一番右下のマスを(6,6)とした場合、(2,4)と(3,5)に居る2つの赤いユニットが自軍のユニットであり、(4,3)と(6,1)に居る2つの青いユニットが敵軍のユニットである。全てのユニットのステータスは同一で、半径3マス以内を移動できる。味方ユニットの存在するマスは通過できるが、移動先にはできない。敵ユニットの存在するマスは通過することも移動先にすることもできない。移動後、隣のマスに敵ユニットが存在しない場合は、何もせず待機をする。移動後、隣のマスに敵ユニットが存在する場合には、その敵ユニットに攻撃して4ダメージを与える。簡略化のため、攻撃可能な場合には必ず攻撃をするものとする。

以上の条件で、自軍ユニットを動かすAIを設計する。一つ一つの移動先や行動に対して、ファジー推論を用いて望ましさを計算し、その数値を影響マップに入れていく。今回の例では、(3,3)のマスに入力する評価値を計算する場合について考える。(3,3)のマスの隣には敵ユニットが存在するので、攻撃して4ダメージ与えることができる。そのため与えるダメージのメンバシップ度は「良い」が0.5、「普通」が0.5となる。また、このマスは味方ユニット2体、敵ユニット1体の攻撃範囲内なので、影響度は4となる。そのため影響度のメンバシップ度は「良い」が0.8、「普通」が0.2となる。図3.7と図3.8は、与えるダメージ4と影響度4そ

それぞれのメンバシップ度を表している。

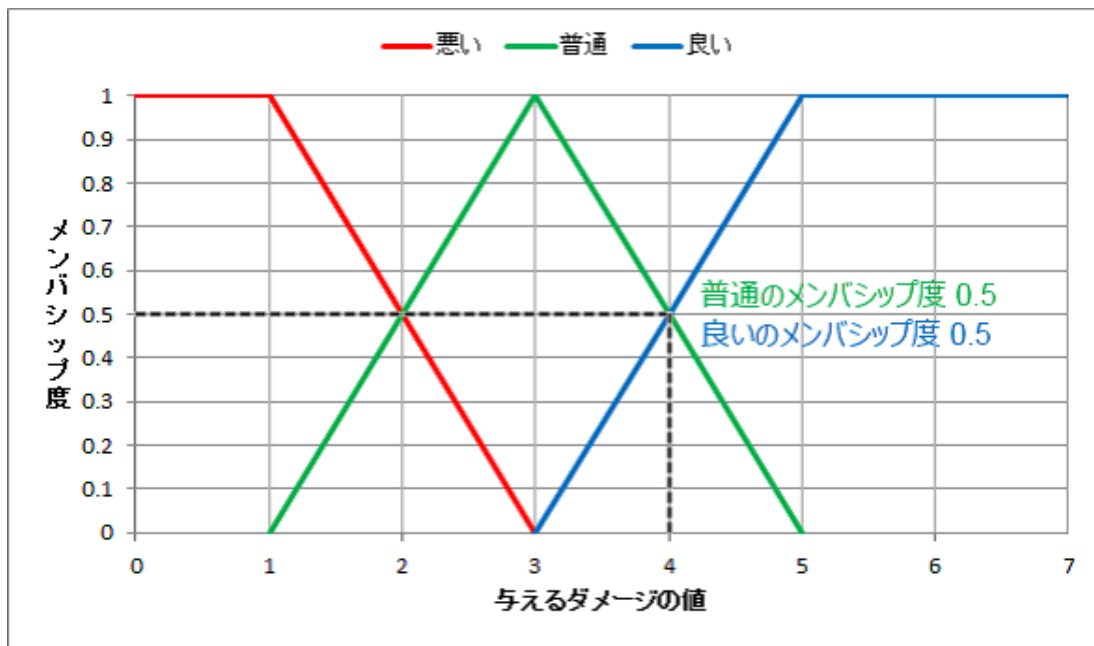


図 3.7: 与えるダメージ 4 のメンバシップ度

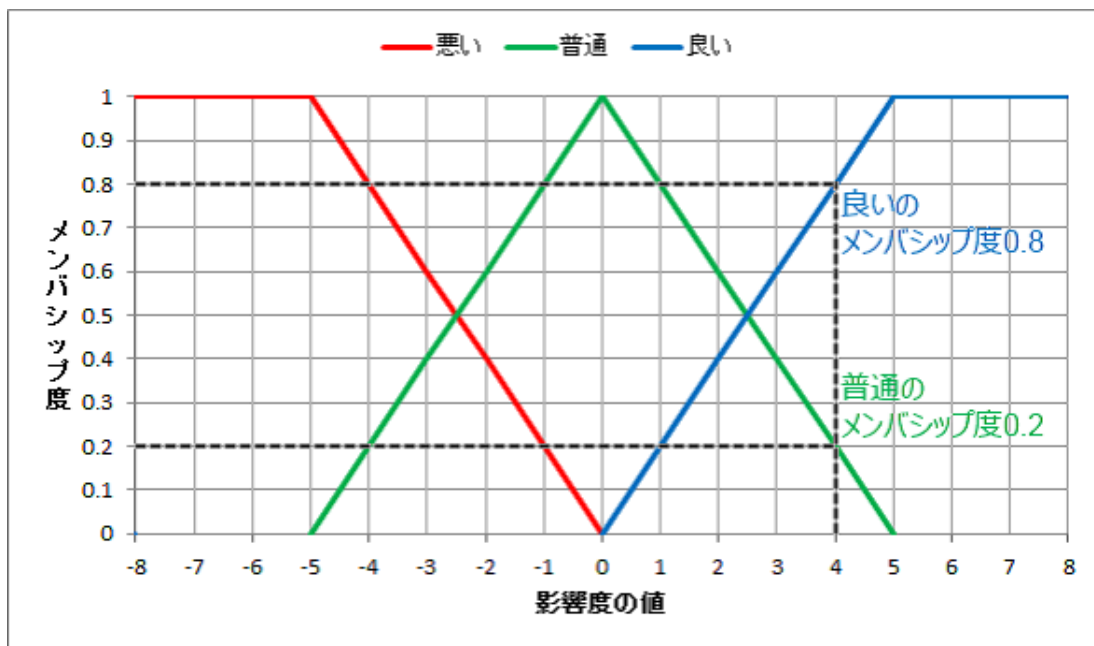


図 3.8: 影響度 4 のメンバシップ度

メンバシップ度の値から、6つのファジールールがどの程度適合するかを計算する。前件部のメンバシップ度がそのまま適合度となり、後件部のメンバシップ度

になる。ルール1の前件部は、与えるダメージが「良い」度合いであり、与えるダメージの「良い」のメンバシップ度は0.5なので、ルール1での推論の結論は、望ましさの「良い」度合いが0.5ということになる。このように計算していくと、各ルールの適合度は次の表3.3の通りになる。

表 3.3: 各ルールの適合度

ルール	前件部	後件部	適合度
ルール1	与えるダメージが「良い」	望ましさは「良い」	0.5
ルール2	与えるダメージが「普通」	望ましさは「普通」	0.5
ルール3	与えるダメージが「悪い」	望ましさは「悪い」	0.0
ルール4	影響度が「良い」	望ましさは「良い」	0.8
ルール5	影響度が「普通」	望ましさは「普通」	0.2
ルール6	影響度が「悪い」	望ましさは「悪い」	0.0

つまり、望ましさ「良い」はそれぞれ適合度 0.8、適合度 0.5 で2つのルールに適合しており、望ましさ「普通」はそれぞれ適合度 0.5、適合度 0.2 で2つのルールに適合していることを意味する。2つ以上のルールに適合している場合は、OR演算を行って最大値を取る。そのため、「良い」の適合度は0.8、「普通」の適合度は0.5となる。与えるダメージ4、影響度4をファジールールに基づいて推論した結果をまとめると、それぞれの適合度は望ましさ「良い」が0.8、望ましさ「普通」が0.5、望ましさ「悪い」が0.0となる。図3.9は、推論結果を合成して生成した、望ましさを表すメンバシップ関数である。

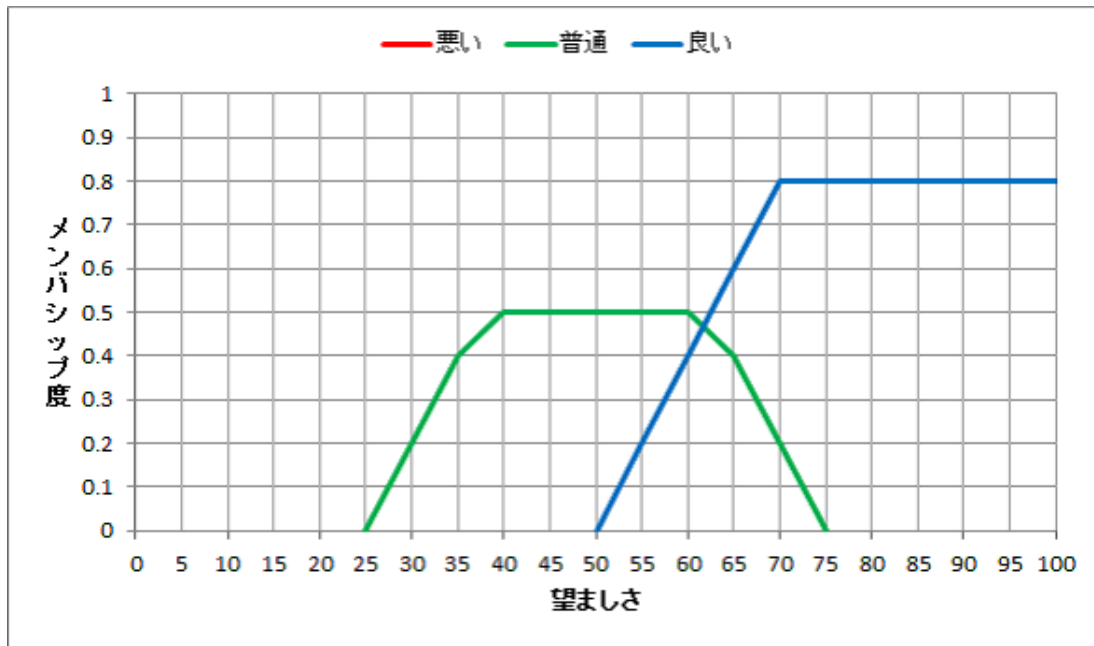


図 3.9: ファジー推論後の望ましさのメンバーシップ関数

最後に、望ましさのメンバーシップ関数から、最終的な望ましさのクリस्प値を出力する。このクリस्प値が、影響マップに入力する評価値となる。メンバーシップ関数の各集合はいびつな形をしているが、これらを統合した1つの図形として考え、重心を計算することで、評価値を出力することができる。メンバーシップ関数の横軸の値を  $x$ 、 $x$  におけるメンバーシップ度の合計を  $f(x)$ 、横軸の最小値を  $a$ 、横軸の最大値を  $b$  とした際、重心  $X$  は次の式 (3.1) のように求める。

$$X = \frac{\int_a^b x f(x) dx}{\int_a^b f(x) dx}. \quad (3.1)$$

この計算を行った結果を、次の表 3.4 にまとめる。

表 3.4: 重心計算

横軸の値	悪い	普通	良い	合計	横軸の値と合計をかけた数
5	0	0	0	0	0
10	0	0	0	0	0
15	0	0	0	0	0
20	0	0	0	0	0
25	0	0	0	0	0
30	0	0.2	0	0.2	6
35	0	0.4	0	0.4	14
40	0	0.5	0	0.5	20
45	0	0.5	0	0.5	22.5
50	0	0.5	0	0.5	25
55	0	0.5	0.2	0.7	38.5
60	0	0.5	0.4	0.9	54
65	0	0.4	0.6	1	65
70	0	0.2	0.8	1	70
75	0	0	0.8	0.8	60
80	0	0	0.8	0.8	64
85	0	0	0.8	0.8	68
90	0	0	0.8	0.8	72
95	0	0	0.8	0.8	76
100	0	0	0.8	0.8	80

分子は横軸の値と合計をかけた数の総和で、735になる。分母はメンバシップ度の合計の総和で、10.5になる。これを計算すると、最終的な望ましさは70となるので、影響マップに70の値を入れる。図 3.10 は、計算結果を影響マップに入力した例である。



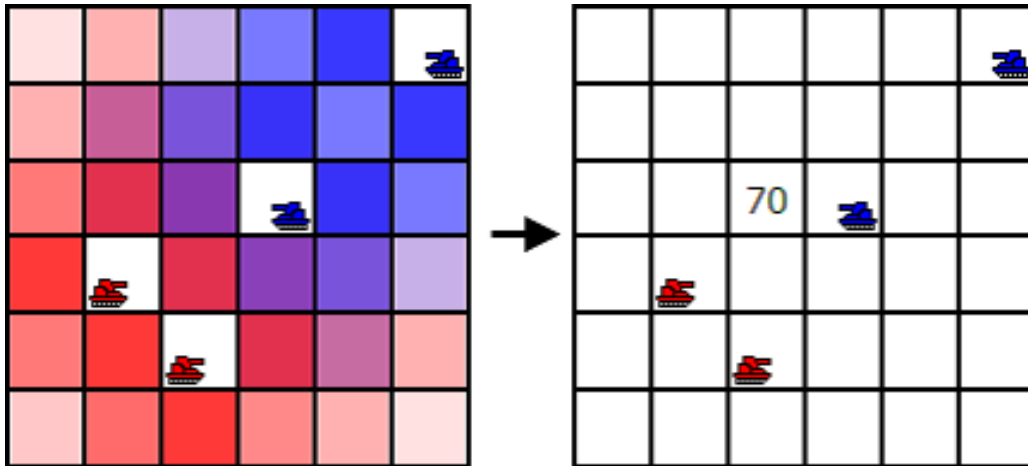


図 3.10: ファジー推論の結果を影響マップに入力した例

このファジー推論の手順を、全てのマスの望ましさを入力するまで繰り返す。全てのマスに望ましさを入力し終わったら、AIは最も望ましさが高い行動を行う。この手法では、ゲーム中で用いるダメージなどの評価項目について、設計者が考えるよし悪しの基準を決め、それらが望ましいかどうかをファジールールで記述することで、最終的な望ましさを自動的に計算して影響マップの生成ができる。設計者が持っている「こういう時はこう判断してほしい」という曖昧な方向性を使って個性的なAIの設計が行えるので、直感的な調整が可能になり手間と労力を大きく削減することができる。

# 第 4 章

## 検証と考察

### 4.1 検証で用いるゲーム環境の解説

本研究の検証に用いる基盤となるゲーム環境である「TUBSTAP」について解説する。ターン制ストラテジーとは、プレイヤーと対戦相手が順番に駒を操作し、目的の達成を目指すゲームのことを指すが、一口にターン制ストラテジーと言っても、ゲームごとに様々な独自のルールやシステムが存在する。例えば「ファミコンウォーズシリーズ」には、都市や工場を制圧して資金を貯め、新たなユニットを生産するというシステムがある。「ファイアーエムブレムシリーズ」や「Final Fantasy Tactics」には、キャラクターを成長させていくというシステムがあり、こういったタイプのゲームはシミュレーションRPGとも言う。このように、ゲームの種類ごとに様々なルールが存在するため、統一されたベンチマークでのAI研究が難しい。

藤木ら [19] は、上記の問題を解決するため、ターン制ストラテジーが持つ様々なルールを整理し、その中から多くのゲームが共通して持っているルールを抽出して、学術用のベンチマークとして使えるターン制ストラテジーのルールを提案している。藤木ら [20] は、提案ルールのゲームを「TUBSTAP」という名前で公開しており、このゲームはターン制ストラテジーのAI研究に用いることができる。次の図 4.1 は、TUBSTAP の実際のゲーム画面である。

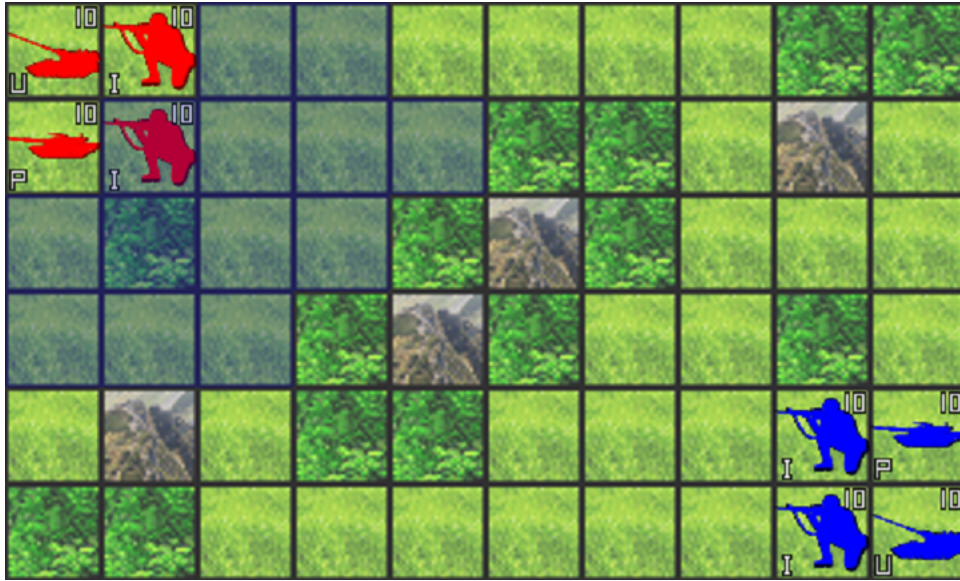


図 4.1: TUBSTAP のゲーム画面

<http://www.jaist.ac.jp/is/labs/ikeda-lab/tbs/>

TUBSTAP は、自分のターン内で複数のユニットを動かすことができる「複数着手性」をはじめ、ユニット同士の戦いで HP を減らしあい、敵を全滅させたら勝利となるなど、多くのターン制ストラテジーに共通する基本的なルールを持ったゲームとなっている。そのため、多くのターン制ストラテジーに応用しやすい研究結果が入手できることが期待できる。また、ルールや GUI、AI 設計に必要なクラスが用意されているため、利用者は AI 設計にのみ専念できるという利点もある。これらの点を踏まえ、本研究の検証実験は TUBSTAP を使用して行うこととする。

## 4.2 実験概要

本研究の目的は、提案手法での AI 設計が、影響マップのみを用いた AI 設計手法（以下、既存手法）よりも、手間と労力を削減できると示すことである。そこで、あらかじめ設定したいいくつかの挙動の AI を、提案手法と既存手法の両方で設計し、目的の挙動を実現するまでに費やしたテストプレイ回数を比較する実験を行う。提案手法で費やしたテストプレイ回数が、既存手法でのテストプレイ回数

を下回れば、提案手法の有効性が示せる。検証用のマップは下の図 4.2 の 3 種類を用いる。

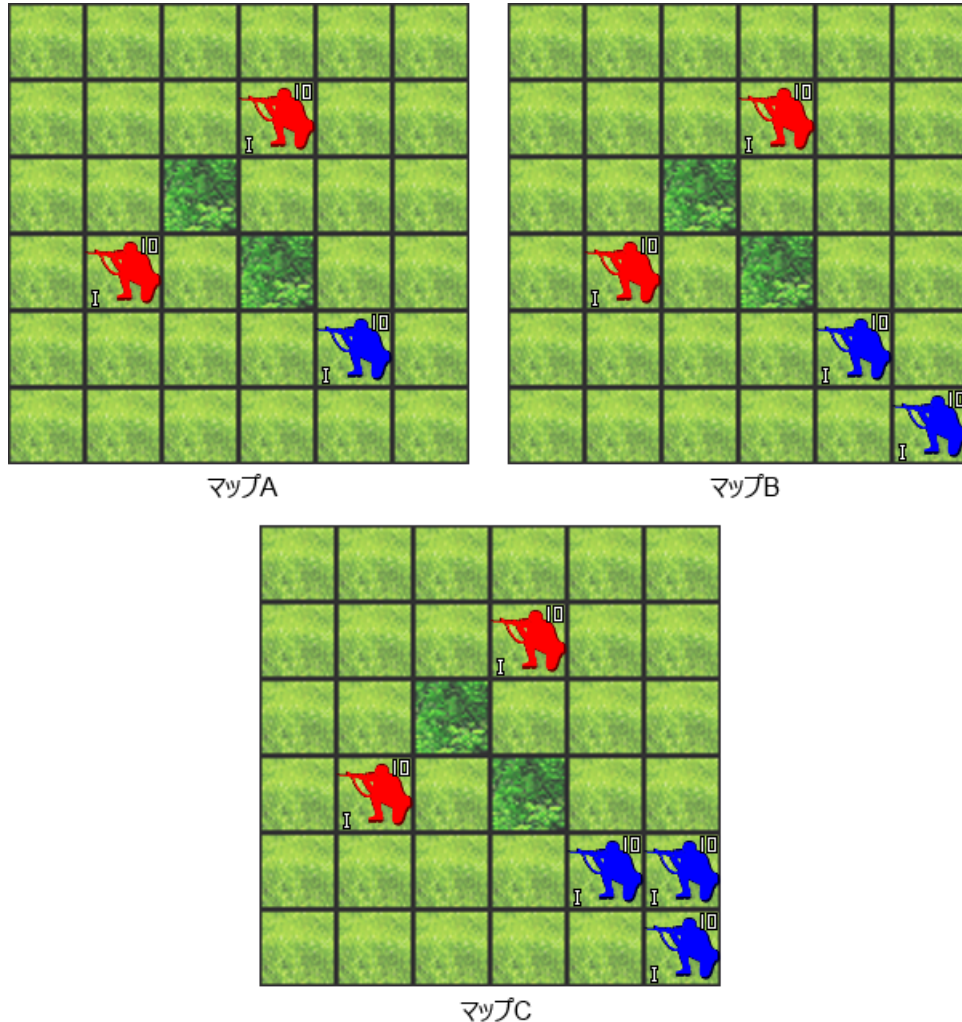


図 4.2: 検証用マップ

いずれのマップも、正方形のマスが縦横それぞれ 6 マス敷き詰められている。一番左上のマスをも (1,1)、一番右下のマスをも (6,6) とした場合、(2,4) と (4,2) に居る赤いユニットが自軍ユニットである。敵ユニットの配置はマップごとに異なる。マップ A では (5,5) の 1 体が存在し、マップ B では (5,5) と (6,6) の 2 体が存在し、マップ C では (5,5) と (6,6) と (6,5) の 3 体が存在する。(3,3) と (4,4) の濃い緑色のマスは森で、上に居る間防御力が 1 上昇する効果がある。

ユニットの種類はすべて歩兵で、HP は 10、攻撃力は 5、移動力は 3 である。HP

が減っていると、攻撃力も減少する。敵ユニットの隣へ移動した際は攻撃が可能で、攻撃力ぶんのダメージを与えることができる。攻撃後、敵ユニットのHPが残っていた場合、反撃を受けるが、先制攻撃をしたぶん敵ユニットのHPは減っているので、受けるダメージは敵ユニットより少なくなる。具体的には、HP10の歩兵でHP10の敵歩兵に攻撃した場合、敵歩兵に5のダメージを与え、こちらの歩兵は3ダメージを受ける。それぞれのAIが考慮する評価項目は、次の3つとする。

- 戦闘結果：攻撃時の、(与えるダメージ) - (反撃で受けるダメージ)
- 影響力：(そのマスに攻撃可能な全ての味方ユニットの攻撃力の和) - (そのマスに攻撃可能なすべての相手ユニットの攻撃力の和)
- 地形効果：地形の防御力修正

全ての移動できるマスに対して、これらの評価項目を元に評価値を入れていく。提案手法の場合は、3つの評価項目の値をファジー集合で表し、ファジールールをもとに望ましさを計算し、影響マップに入力していく。1つの評価項目が持つ集合は、「非常に良い」「良い」「普通」「悪い」「非常に悪い」の5つとする。既存手法の場合は、それぞれの評価項目が出力した数値に、修正値を足したり引いたりして調整を行う。生成するAIの挙動は、基準AI・防御的AI・地形重視AIの3種類とする。基準AIは有利な際は攻撃し不利な際は逃げるAIである。防御的AIは基準AIに比べ、被害を受けないことを重視する。地形重視AIは基準AIに比べ、森の上へ移動することを重視する。それぞれのAIの具体的な行動は、次の通りである。

- 基準AI：マップA・マップBでは、(5,5)の歩兵に両歩兵で攻撃を仕掛ける。マップCでは敵が多く不利なため、相手の攻撃範囲外に逃げる。
- 防御的AI：マップAでは、(5,5)の歩兵に両歩兵で攻撃を仕掛ける。マップB・マップCでは、相手の攻撃範囲外に逃げる。

- 地形重視 AI：マップ A・マップ B では、(5,5) の歩兵に両歩兵で攻撃を仕掛ける。マップ C では、1 体の歩兵を (3,3) の森へ待機させ、もう 1 体の歩兵は相手の攻撃範囲外に逃げる。

1 つの AI について、マップ A・マップ B・マップ C で同時に条件を満たした時、AI の設計が実現できたものとする。提案手法と既存手法でそれぞれ AI 設計を行い、AI の実現にかかったテストプレイ回数を競った。なお、変数名を誤ったなどの別の要因でのミスがあった場合は、その分のテストプレイは回数としてカウントしない。

### 4.3 実験結果

3 種類の AI の挙動を、提案手法と既存手法でそれぞれ設計した結果、目的の挙動を実現した時点でのテストプレイ回数は次の通りとなった。

- 基準 AI：提案手法...2 回      既存手法...3 回
- 防御的 AI：提案手法...1 回      既存手法...2 回
- 地形重視 AI：提案手法...2 回      既存手法...9 回

それぞれの AI でどのような設計を行ったかについて述べる。1 番目の基準 AI は、挙動が単純なためどちらの手法でも少ないテストプレイ回数で実現した。提案手法における各ファジー集合の基準は、次の表 4.1 のようになった。

表 4.1: 基準 AI における提案手法でのファジー集合の基準

評価項目	非常に良い	良い	普通	悪い	非常に悪い
戦闘結果	4	2	0	-2	-4
影響力	10	5	0	-5	-10
地形効果	4	2	0	-	-

戦闘結果については、HP10 の歩兵で HP10 の敵歩兵に攻撃した際、敵歩兵に 5 ダメージ与え、こちらの歩兵は 3 ダメージを受け、2 ダメージぶん優位に立つこと

になるので、2を「良い」に設定し、これを基準として他の集合の値も設定した。影響力については、HP10の歩兵の数が相手よりも1体上回っている際、攻撃力5ぶん優位に立つことになるので、5を「良い」に設定し、これを基準として他の集合の値も設定した。地形効果については、TUBSTAPでの地形効果の最小値が0、最大値が4であるため、4に近づくとつれ「非常に良い」に近くなるように設定した。ファジールールは次の表4.2のように記述した。

表 4.2: 基準 AI におけるファジールール

ルール	前件部	後件部
ルール1	戦闘結果が「非常に良い」	望ましさは「非常に良い」
ルール2	戦闘結果が「良い」	望ましさは「良い」
ルール3	戦闘結果が「普通」	望ましさは「普通」
ルール4	戦闘結果が「悪い」	望ましさは「非常に悪い」
ルール5	戦闘結果が「非常に悪い」	望ましさは「非常に悪い」
ルール6	影響力が「非常に良い」	望ましさは「非常に良い」
ルール7	影響力が「良い」	望ましさは「良い」
ルール8	影響力が「普通」	望ましさは「普通」
ルール9	影響力が「悪い」	望ましさは「悪い」
ルール10	影響力が「非常に悪い」	望ましさは「非常に悪い」
ルール11	地形効果が「非常に良い」	望ましさは「良い」
ルール12	地形効果が「良い」	望ましさは「良い」
ルール13	地形効果が「普通」	望ましさは「普通」

ルール4で、戦闘結果が「悪い」ことで望ましさが「非常に悪い」としているのは、被害が大きくなる攻撃を自分から仕掛けるのが非常に不利になるとするためである。ルール11で、地形効果が「非常に良い」ことで望ましさが「良い」としているのは、地形効果はあくまで補助の評価項目とするためである。このようなファジールールの設定を行うことに、2回のテストプレイを費やした。対して既存手法では、戦闘結果の影響マップには「(与えるダメージ) - (反撃で受けるダメージ)」に3を加えた値を入れ、影響力の影響マップには「(味方ユニットの攻撃力の和) - (相手ユニットの攻撃力の和)」の値をそのまま入れ、地形効果の影響マップには、森のマスに1を入れた。これら3つの影響マップを足しあわせ、最

終的な影響マップとした。

2番目の防御的 AI は、どちらの手法においても基準 AI を少し修正して設計した。提案手法では、ファジールールはそのままに、戦闘結果のファジー集合の基準を次の表 4.3 のように変更することで実現した。基準 AI よりも、攻撃を慎重にしたいという AI なので、戦闘結果に対する評価を 1 だけ厳しい方向に調整することで、1 度のテストプレイで実現した。

表 4.3: 防御的 AI で変更したファジー集合の基準

評価項目	非常に良い	良い	普通	悪い	非常に悪い
戦闘結果	5	3	1	-1	-3
影響力	10	5	0	-5	-10
地形効果	4	2	0	-	-

既存手法では、戦闘結果の影響マップには「(与えるダメージ) - (反撃で受けるダメージ)」から 1 を引いた値を入れ、影響力の影響マップには「(味方ユニットの攻撃力の和) - (相手ユニットの攻撃力の和)」の値をそのまま入れ、地形効果の影響マップには、森のマスに 1 を入れた。

3番目の地形重視 AI は、有利なときは攻撃を仕掛け、そうでない時は (3,3) の森へ移動し、相手の集中攻撃を受けてしまう (4,4) の森には入らないという AI である。地形効果に高い評価をするだけでは (4,4) の森に入ってしまうため、前の 2 つの AI よりも複雑であると言える。提案手法では、ファジールールはそのままに、地形効果のファジー集合の基準を次の表 4.4 のように変更した。戦闘結果に対する評価基準を標準 AI のものに戻し、地形効果に対する評価を緩くすることで、2 回のテストプレイで実現できた。

表 4.4: 地形重視 AI における提案手法でのファジー集合の基準

ファジー変数	非常に良い	良い	普通	悪い	非常に悪い
戦闘結果	4	2	0	-2	-4
影響力	10	5	0	-5	-10
地形効果	3	1	0	-	-



既存手法では、戦闘結果の影響マップには「(与えるダメージ) - (反撃で受けるダメージ)」に3を加えた値を入れ、影響力の影響マップには「(味方ユニットの攻撃力の和) - (相手ユニットの攻撃力の和)」の値をそのまま入れ、地形効果の影響マップには、森のマスに7を入れた。森へは移動するがもう1体が攻撃してしまう、マップAやマップBでも森に入ってしまうなどの不具合の解消に時間がかかった。

#### 4.4 考察

3つのAIを設計する実験を行ったところ、全てのAIにおいて提案手法のテストプレイ回数が既存手法のテストプレイ回数を下回るという結果になった。単純なAIはどちらの手法でもすぐに実装ができるものの、地形重視AIのような複雑な挙動をするAIを設計する際には、既存手法では入力すべき値の目安が直感的に分からず、時間がかかった。それに対し提案手法では、ゲーム中で扱う数値の良し悪しの基準を設定することで影響マップが生成できるので、どの項目をどれくらい調整すべきなのかが直感的に分かり、試行錯誤の手間と労力を減らすことができた。

## 第 5 章

### まとめ

本研究では、個性的な AI の設計にかかる手間と労力を削減するため、影響マップとファジー論理を組み合わせた手法を提案し、既存手法と比較検証を行った。その結果、提案手法では評価の良し悪しの基準とファジールールの直感的な調整により AI 設計が行え、影響マップのみを用いた手法よりも手間と労力が削減できることがわかった。数値で評価可能な評価項目ならファジー論理で表すことができるので、様々な個性を持った AI の設計に用いることができ、個性的な AI 設計がより直感的に可能になると言える。今回の実験では、狭いマップで限定された状況下での実験を行ったが、今後はもっと広いマップや、多くの地形・ユニットが存在する状況において、多彩な個性を持つ AI 設計の実験が必要であると考えられる。

# 謝辞

本研究を進めるにあたり、多大なご指導をして頂いた渡辺先生、三上先生、阿部先生に心より感謝申し上げます。なかなか研究テーマが決まらない中、相談に乗って頂いた院生の方々や、共に研究を進めてきた研究室のメンバーにも感謝致します。ありがとうございました。

## 参考文献

- [1] ファミコンウォーズ DS. <http://www.nintendo.co.jp/ds/awrj/>. 参照：2015-02-21.
- [2] Civilization シリーズ. <https://www.civilization.com/jp/home/>. 参照：2015-02-07.
- [3] 上田陽平, 池田心. 遺伝的アルゴリズムによる人間のレベルに適応する多様なオセロ ai の生成. 研究報告ゲーム情報学 (GI), 2012.
- [4] 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. GI, [ゲーム情報学], 2010.
- [5] 仲道隆史, 伊藤毅志. 機械学習を用いた棋力の調整方法の提案と認知科学的評価. 情報処理学会研究報告. GI, [ゲーム情報学], 2013.
- [6] 志水翔, 金子知適. 二人ゲームプレイヤーの prior knowledge を用いた uct による個性の実現手法と評価. ゲームプログラミングワークショップ 2014 論文集, 2014.
- [7] 加藤千裕, 三輪誠, 鶴岡慶雅, 近山隆. ターン制ストラテジーゲームにおける戦術決定のための uct 探索とその効率化. ゲームプログラミングワークショップ 2013 論文集, 2013.

- [8] 藤木翼, 村山公志朗, 池田心. ターン制ストラテジーのための状態評価型深さ限定モンテカルロ法における消極的行動の抑制. ゲームプログラミングワークショップ 2014 論文集, 2014.
- [9] 三宅陽一郎. オンラインゲームにおける人工知能・プロシージャル技術の応用. 日本知能情報ファジィ学会誌, 2010.
- [10] 株式会社モバイル&ゲームスタジオ. 遠藤雅伸のゲームデザイン講義実況中継. SBクリエイティブ株式会社, 2014.
- [11] 狩野 智英・鳥海 有紀 MARK DELOURA. *GAME PROGRAMMING Gems* 2. 株式会社ボーンデジタル, 2002.
- [12] 三宅陽一郎. GDC における海外のゲーム関連技術についての調査. 2010.
- [13] 橋口ゆうすけ. ゲームのアルゴリズム改訂版 思考ルーチンと物理シミュレーション. ソフトバンククリエイティブ, 2012.
- [14] Glenn Seemann David M.Bourg. ゲーム開発者のための AI 入門. オライリー・ジャパン, 2005.
- [15] 狩野 智英 MARK DELOURA. *GAME PROGRAMMING Gems*. 株式会社ボーンデジタル, 2001.
- [16] 亀井 且有星野 孝総. ファジィ環境評価ルールを用いた強化学習の提案とチェスへの応用. 日本ファジィ学会誌, 2001.
- [17] 大西克哉. ファジィ理論を用いたゲームバランス調整作業の効率化. 東京工科大学, 2008.
- [18] 松田 晃一 Mat Buckland. 実例で学ぶゲーム AI プログラミング. オライリー・ジャパン, 2007.

- [19] 村山公志朗, 藤木翼, 池田心. 学術研究用プラットフォームとしての大戦略系ゲームのルール提案. ゲームプログラミングワークショップ 2013 論文集, 2013.
- [20] ターン制戦略ゲーム 学術用基盤プロジェクト TUBSTAP.  
<http://www.jaist.ac.jp/is/labs/ikeda-lab/tbs/>. 参照 : 2015-02-07.