情報処理学会論文誌 Vol.57 No.11 0–16 (Nov. 2016)

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

佐藤 直之[1,a)] 藤木 翼[1,b)] 池田 心[1,c)]

受付日 2016年6月1日, 採録日 2016年8月8日

概要：本稿は「戦術的ターン制ストラテジー」という，チェスや将棋と似た形式でアプローチしやすく，また同時に３つの興味深い課題を含むAI設計の問題クラスを記述する．その課題とは，１つ目は行動数の組合せ爆発で，同ゲームでは１手番ごとのbranching factorがしばしば億のオーダーに達する．２つ目は局面評価に関するもので，毎回異なる初期局面から生じる多様な局面群に対し，駒間の循環的相性も考慮して駒価値を適切に与えなければならない．３つ目は攻撃行動組合せの扱いが要する繊細さで，同ゲームでは攻撃行動の適切な組み合わせで数十体の駒ものがたった一手番で消滅する事があり，そうした影響力の行使および相手からの行使の予防が重要になる．我々はこれらの課題を，具体的状況と既存のAI手法を例に用いて論じた．複数のアプローチを提案しそれぞれの長所と短所を整理して，同問題においてAI設計者が考慮すべき課題の特徴を明らかにした．

キーワード：ゲームAI，ターン制戦略ゲーム，ターン制ストラテジー，モンテカルロ木探索

Proposal of Challenges and Approaches to Create Effective Artificial Players for Turn-based Tactics Game

Naoyuki Sato[1,a)] Fujiki Tsubasa[1,b)] Kokolo Ikeda[1,c)]

Received: June 1, 2016, Accepted: August 8, 2016

Abstract: This paper describes characteristics and problems with designing AI players in “Turn-based tactics” games. These environments of these games provide the designers a similar framework of designing AI players while these provide them some interesting challenges to deal with three major problems described below. Firstly, branching factors of the game tree search often exceed hundreds millions in the games. Secondly, the evaluation of game positions is often difficult in the game because the effectiveness of pieces varies drastically according to the types of opponent pieces in the games. Thirdly, combinations of attack actions in the games have a potentially great effect on game situations. We discussed the effects made by these problems in detail suggesting multiple approaches for the problems, moreover we discussed about the dis/advantages in each approach in example situations. Finally, we made the characteristics of the problems that AI designers face with in the game.

Keywords: Game AI, Turn-based Strategy game, Turn-based Tactics game, Monte-Carlo Tree Search

1. はじめに

人工知能研究の一分野として，ゲームにおけるコンピュータプレイヤ（以下，AIプレイヤと呼ぶ）の作成は古くから研究されてきた．研究の目的は様々あるが，特にそれぞれのゲームにおいて「強いAIプレイヤの開発」は主要なテーマの１つである．そうして作られたAIプレイヤの強さは，例えばチェスや将棋，あるいは囲碁で目覚ましい．チェスではIBM社の開発したDeepBlueが当時の世界チャンピオンに勝利していて[1]，将棋と囲碁でも人間のプロに匹敵する強さのプログラムの開発に成功している[2][3]．つま

[1] 北陸先端科学技術大学院大学
JAIST, Nomi, Ishikawa 923–1211, Japan
[a)] satonao@jaist.ac.jp
[b)] s1310062@jaist.ac.jp
[c)] kokolo@jaist.ac.jp

りこれらのゲームでAIプレイヤは一般的な人間プレイヤの相手を務めるために十分な強さを持っている．

一方でルールがより複雑なゲームの中には，AIプレイヤの強さが人間の上級者に対して十分ではないものも多くあり，そうしたゲームでは更なる研究が望まれる．例えば不完全情報ゲームの麻雀 [4] や，リアルタイム性でゲームが進行するStarCraft[8] などである．そして一ターンに複数の駒を操作できるターン制ストラテジーゲームもその1つである．このジャンルのゲームは人気が高く，累計100万本以上の売り上げを持つ大戦略シリーズ [6] や300万本以上の売り上げのCivilizationシリーズ [7] など様々なタイトルが広く遊ばれている．

一口にターン制ストラテジーゲームといっても様々な種類があって，プレイヤに必要な考え方も変わってくる．特に「内政」のルールなど政治的要素をゲームに含むかどうかはプレイヤの考え方を大きく分ける．Civilizationなどの，内政があるゲームではプレイヤは概して政治的な行動ののちに他プレイヤとの戦闘に移る．その政治的な部分での優れた立ち回りにより相手より多くの戦力（多くの強い駒）を用意して，自分に有利な条件で戦闘を始めることを目指す．

一方で大戦略シリーズなど内政が無いタイプではゲームが戦闘から始まる．互いの初期戦力は事前に設計者に決められていて，戦闘のシステム自体も前述のタイプより複雑に設計されていることが多い [10]．そのため勝敗の決定には駒の用兵が占める比重が高く，プレイヤは戦闘で駒の移動先を1マス単位で気にしたり，各駒の行動順や攻撃対象を注意深く選択する必要に迫られる．

ただしターン制ストラテジーの既存研究は内政のあるタイプに関するものが多く [11]，内政のないタイプのゲームに関してはかなり少ない．

しかし内政のないターン制ストラテジーは，研究対象として見るとチェスや将棋とある程度形式が近く，それら古典的ボードゲームで従来使われてきた木探索手法によるアプローチがしやすい．その一方で強いAIを作るためには，対処に工夫を要する性質もいくつか含まれている．例えば，手番ごとの「合法手の組合せの多さ」や，形勢判断時の「駒価値の変動」などである．こうした要素は現状で多くの人間プレイヤが感覚的に対応出来ているものであるため，AIによるアプローチは興味深い．

そこで我々はAI設計問題の対象として内政のないターン制ストラテジーに注目したい．このジャンルはゲームタイトル間に様々なルールのバラつきがあるため研究対象として統一的に扱う事は難しい．また各タイトルは大抵，他のタイトルのいずれかには含まれないルールを1つ以上持っている．そのため特定の既存タイトルを研究対象に選ぶとそのタイトルに特化した問題を扱う事になってしまうと我々は考える．

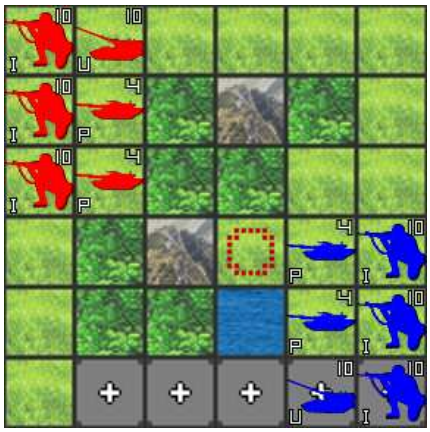

図 1 ターン制ストラテジーの例．プラットフォーム『TUBSTAP』スクリーンショット．手番ごとにマスの上の駒を動かして，敵の駒を攻撃していく．

Fig. 1 An example of turn based tactics, “TUBSTAP” platform. Each player manipulate their pieces to fight against each other.

そのため我々は内政のないターン制ストラテジーを全般的に扱うための入り口となる問題クラスを提案し，AI設計問題の対象として提示したい．この問題クラスは，各タイトルに共通して含まれるルールを切り出す事で，内政のないターン制ストラテジー全般に共通する問題だけを集中的に扱う事を可能にする．またそうした限定的な切り出しによって，問題クラスを入り口として取り組むのにふさわしい難易度に留めると我々は考える．

我々はそうした，内政のないターン制ストラテジーでのAI設計の入り口として適した問題クラスと（そのクラスでのAI設計時に）考えるべき課題の紹介を本稿の目的とする．まず内政のないターン制ストラテジーは内部に様々なルールのバラつきがあるため，我々はゲームのルールに適切な絞り込みを与えてAI設計問題を設定した．こうして設定されたゲームクラスを我々は便宜的に「戦術的TBS」と呼ぶ（TBSはTurn-Based Strategyの頭文字である）．本稿は，その戦術的TBSでAIを設計する際にルールがAI設計に与える影響と具体例を3個紹介してゲームクラスでのAI設計が持つ課題を明確にする．

本稿の構成は以下である．まず戦術的TBSとその関連するゲームジャンルとの区別を2章で与える．そして，その戦術的TBSの分類の中にもゲームタイトルごとにルールのバラつきが見られるため，本稿が着目する問題領域の切り抜きを3章で行う．4章では戦術的TBSへの適用が考えられる主な既存ゲームAI手法を列挙する．5章からは戦術的TBSでのAI作成が持つ3つの課題を具体例とともに解説し，8章がまとめである．

2. ターン制ストラテジーと他ゲームの関連

ターン制ストラテジーは図1のようにそれぞれのプレイ

ヤーが交互に盤上の駒を動かす事で相手の陣営と戦うゲームであり，基本的にはチェスや将棋などの古典的ボードゲームに似ている．ただし内政ルールのあるタイプとないタイプでゲームの性質はかなり変わり，またターン制ストラテジー自体が他のいくつかのゲームジャンルと関連もする．それらの区別には紛らわしい部分もあるため本章で整理する．

2.1 ターン制ストラテジー

一般にターン制ストラテジーは手番交代制の戦略シミュレーション（戦争を模したゲーム）のジャンルであり，古典的ボードゲームと比べた主なルールの違いは以下の形で記述できる．これら５つの性質はゲームタイトルに依らずほぼ全てのターン制ストラテジーに共通する要素である．

- **複数着手性**：単一の駒しか一度に動かせないチェス等と異なり，手番ごとに複数の駒を任意の順序で動かせるルール．
- **駒の攻撃**：チェス等における，相手の駒の位置に自分の駒を進めることで駒を除去するルールと異なって，相手の駒を除去するためには駒が攻撃という行動を行う．隣接または一定距離離れた敵の駒を対象にする．隣接攻撃の直後には相手からの「反撃」がルールとして伴う場合が多い．
- **HP**：駒の攻撃に関係した指標である．駒の体力・健全度などを示し，敵からの攻撃により減り，0 になると盤から除去される．
- **地形**：チェス等では通常，個別のマスそれぞれはほとんど個性を持たないのに対し，ターン制ストラテジーでは各マスに沼地・要塞など様々な地形が割り当てられており移動や攻撃に影響を与える．
- **初期局面多様性**：常に同一の局面からゲームが始まるチェス等と違って，各ゲームが設計者によって作られた局面から始まる．

さらに，ターン制ストラテジーはタイトルによっては例えば以下のようなルール要素を備える場合がある．

- **駒の相性**：じゃんけんのような駒の間の得手不得手の関係で，ある駒からある駒への攻撃による HP 減少値に偏りを与える．
- **ランダム性**：攻撃による HP 減少値が確率的に上下する．また，攻撃自体が失敗してしまうこともある．
- **占領**：工場や都市など，ある特殊な地形マスに「歩兵」などの駒を移動させることでマスの所有権を得る．
- **生産**：駒を新たに製造できる特殊なマス（例えば工場）を所有している場合，手番ごとに「資金」（都市マスなどの所持により増加）などのパラメータを消費して自陣の駒を新たに増やせる．
- **ZoC**：駒が自分の周辺のマスに Zone of Control というエリアを持ち，そこを通過しようとする敵駒の移動可能範囲を大きく狭めるルール．
- **範囲攻撃**：ある一定の区域にいる敵の駒全てに一度に攻撃を加えることができるルール．
- **索敵**：自陣の駒の限られた周囲の領域しか敵駒の存在が確認できないシステム．ゲームに不完全情報性をもたらす．
- **地形の多層性**：１マスの地形が，地上と空，海底と海上と空など多層の構造からできている盤の使用．

そしてゲームの性質を大きく変えると我々が考えるルールは以下である．

- **内政**：ある政治的な特殊行動によって生産力（生産できる駒の質と量の度合い）を向上させる．または，特殊マスを盤上に新たに追加したり，索敵で敵駒が発見できる範囲を広げられたりする．

この内政ルールの有無についてゲームを分類した場合の，それぞれの特徴を次節から説明していく．

2.1.1 内政要素のないクラス

本稿で着目するゲームクラスであり，内政ルールを含まない．大戦略シリーズや，ネクタリスなどのタイトルがこれに含まれる．

内政要素のあるものと違って，政治的な行動のフェーズを経ずに戦いが始まり，プレイヤに与えられる視点も局所的である．またほとんどのタイトルに含まれる「駒の相性」もその効果の度合いが大きく，ごく少数の駒が相性によってかなり多数の駒を一方的に打ち破る事も珍しくない．そうした事情もあって，戦場の細かな用兵の違いが戦果を大きく左右しやすいといった特徴も見られる．

既存研究に関して言えば内政のあるタイプに比べてかなり少ない．例えばファミコンウォーズを模した自作環境を用い，進化計算により最適化されたパラメータを持つ AI 作成を行ったもの [12] や，三国志 IX の戦闘部分だけをモデル化した自作環境上で，行動の枝刈りを伴う UCT 探索を適用したもの [11] がある．

また TUBSTAP と呼ばれるプラットフォームにてシミュレーション深さ限定型のモンテカルロ法利用の木探索 [13] や，ファジィ関数の枝刈り型 UCT 木探索の適用 [14] などが報告されている．さらに，本稿で提案する問題クラスと同一な，内政の無いターン制ストラテジーゲームへの接近法となる問題クラスの提案といくつかの AI の対戦実験結果は既に報告されている [28]．しかし本稿はそれにくわえて，問題クラスが設計者に与える課題３つを明確化する．

2.1.2 内政要素のあるクラス

内政ルールを含むターン制ストラテジーで，Civilization や信長の野望シリーズ [15] が有名である．このゲームでは内政ルールのため，プレイヤはまず政治的な行動によって相手より多くの軍事力を確保しようとする展開になりや

すい．

またプレイヤに与えられる視点も巨視的である．戦闘のシステムにおいては前節のタイプより抽象化と簡略化が見られて [10]，タイトルにもよるが，このジャンルでの細かな用兵の違いは戦果を大きく左右しづらい傾向がある．

このジャンルの既存研究は Civilization シリーズが広く用いられている．都市建設または指導者 AI の切り替えに Q 学習を適用した例 [16][17] や，マニュアルの自然言語から行動評価関数の調整を行った例がある [18]．

また Civilization のクローン環境を利用した研究も多い．都市防衛のタスクを AI に学習させたり [19]，駒の経路選択の学習を行った研究 [20] がある．あるいは将来の資源生産量の推定を行ったり，食糧管理タスクを事例に基づき学習する試み [21] もみられる．

2.2 シミュレーション RPG

SRPG は，ターン制ストラテジーに Role Playing Game（RPG）の要素を盛り込んだ派生型である．ターン制ストラテジーの駒 1 つ 1 つに物語のキャラクタを演じさせ，敵との戦闘を通じてそれらを成長させていくことで従来の RPG に似た楽しみ方をプレイヤに提供する．代表的なタイトルには国内 170 万本の売り上げを達成した Final Fantasy Tactics[5] があり，他にもこのジャンルのゲームは日本国内で多くのタイトルが発売されている．

具体的なルール設定としては前述のターン制ストラテジーのルール要素に加え，以下のルール両方を持つものをこのジャンルとして考える．

- **キャラクタ性**：ほぼすべての駒が，ある 1 つの人格付けされたキャラクタを持つ．駒の性能にもキャラクタの個性が反映されていて，それぞれ異なる．
- **駒の育成**：駒に成長の度合いを表すパラメータ（レベル）があり，各ゲームをまたいで引き継がれる．

また SRPG の AI 設計指針に関しては SRPG は内政要素のないターン制ストラテジーとかなり近い．というのも SRPG では物語のパートと戦闘のパートは独立に扱える事が多く，そして戦闘のパートは内政ルールを含まないターン制ストラテジーによく似ている．

もし違いを挙げるとすれば，SRPG では駒の相性があまり顕著でない一方で，キャラクタとしての駒を成長（経験値を多く与えて戦闘に関する性能を上げる）させることで戦闘開始時の戦力差に偏りを与えられる点がある．また，キャラクタの活躍や敗北が物語に影響を持つこともよくあって，戦闘の最終的な勝利のために自軍の一部の駒を犠牲とする戦略を選べないような場面もしばしば現れる．

2.3 リアルタイムストラテジー

RTS はターン制ストラテジーの基礎的なルールのうち「複数着手性」を持たない代わりに以下のリアルタイム性を持つような，戦略シミュレーションの一つである．

- **リアルタイム性**：各プレイヤが任意のタイミングで駒を操作する．

また，そうした時間の（疑似）連続化にともない，盤上の駒の位置情報も（疑似）連続値的に変動して，ゲームの状態はめまぐるしく変化する．このジャンルでは StarCraft や Age of Empire などのシリーズがタイトルとして有名である．

また，RTS は内政に関するルールを含むことが多く，ゲームの性質としては内政のあるターン制ストラテジーに近い．とはいえリアルタイム性のために，敵の行動に対して素早い反応を求められることもあり，時間をかけてゆっくり戦略を練れるターン制ストラテジーとはゲームに適した AI 設計もかなり違ってくると考えられる．学術研究は盛んに行われており，AI の競技会も定期的に開かれている [8]．既存研究はかなり多くの点数が，広範な種類の部分問題を対象に行われているが，Santiago らがそれらに適切な概観および分類を与えている [9]．

2.4 Arimaa と軍人将棋

やや特殊な例として，Arimaa と軍人将棋を挙げる．まず Arimaa はチェスの盤と駒を利用していて，チェスや将棋等の古典的ボードゲームに色合いがかなり近いが，ターン制ストラテジーに広くみられるような複数着手性も持っている．

そのため AI 設計時には，戦術的 TBS と同様に，ゲーム木の枝の多さが問題となりやすい．ただしそうした共通点の一方で両者のルールには決定的な違いも多く，例えば Arimaa は駒に遠隔攻撃や HP はなく，駒の除去もトラップという特殊なマスで行われる．

軍人将棋も古典的ボードゲームと色合いが近いが，駒に極端な相性があり，初期局面にも多様性があって，ターン制ストラテジーのルール要素をいくつか備える．ただしこれも駒の攻撃や HP は無かったり，複数着手性が無かったり，異なる部分も多い．特にプレイヤが自分の駒を裏返しに伏せたまま操作する事でゲームが不完全情報性を備えており，大きな差である．

2.5 ゲームタイトルごとの表

これまでに挙げたゲームジャンルそれぞれから計 17 のゲームタイトルを選び，ルールの違いを表 1 にて示す．タイトルの選択は恣意的だが，国内でそれなりに著名なものを選んだ．研究用プラットフォーム TUBSTAP，そして Arimaa とチェスも比較のため上記ルール要素の枠組みで載せてある．

F ウォーズは「ファミコンウォーズ」，A 大戦略は「アドバンスド大戦略」，AoE2 は「Age of Empire2」，FFT は「Final Fantasy Tacitics」，T オウガは「タクティクスオウ

表 1 主要な戦略シミュレーションゲームごとのルールの違い：ルール要素への縦 2 重線区切りは，左から「多くに共通するルール」「やや発展的なルール」「稀なルール」の分類
Table 1 Defference among basic rules of representative turn based strategy games.

ジャンル	タイトル	着手	攻撃	HP	地形	相性	生産	占領	索敵	内政	ZOC	多層地形	育成	キャラ性
古典的ボードゲーム	チェス	単	×	×	×	×	×	×	×	×	×	×	×	×
	将棋	単	×	×	×	×	×	×	×	×	×	×	×	×
	Arimaa	複数	×	×	×	×	×	×	×	×	×	×	×	×
	軍人将棋	単	×	×	×	○	×	×	×	×	×	×	×	×
研究環境	TUBSTAP	複数	○	○	○	○	×	×	×	×	×	×	×	×
内政が無いタイプ	ネクタリス	複数	○	○	○	○	×	○	×	×	○	×	×	×
	現代大戦略	複数	○	○	○	○	○	○	×	×	×	×	×	×
	F ウォーズ	複数	○	○	○	○	○	○	×	×	×	×	×	×
	大戦略 EX	複数	○	○	○	○	○	○	○	×	○	○	×	×
	A 大戦略	複数	○	○	○	○	○	○	○	×	○	○	○	×
内政があるタイプ	CivilizationV	複数	○	○	○	×	○	○	○	○	×	×	×	×
	ギレンの野望	複数	○	○	○	○	○	○	○	○	×	×	×	×
	三国志 IX	セミ RT	○	○	○	○	○	○	×	○	×	×	×	×
RTS	AoE 2	RT	○	○	○	○	○	×	○	○	×	×	×	×
	Star Craft 2	RT	○	○	○	○	○	×	○	○	×	×	×	×
SRPG	FFT	素早さ	○	○	○	○	×	×	×	×	×	×	○	○
	T オウガ	素早さ	○	○	○	○	×	×	×	×	×	×	○	○

ガ」をそれぞれ表す．また着手の RT はリアルタイム性，セミ RT はセミリアルタイム性（思考時間中に両プレイヤが行動を決定後，それらが同時に解決），素早さは駒の「素早さ」パラメータの高い順に 1 つずつ行動する着手形態である．

表のように，ある種のルール要素は同一ジャンル内だけでなく，ジャンルをまたいで様々なタイトルに採用されている場合もある．その一方で，同一ジャンルの中でさえも採用される場合とされない場合に分かれるようなルール要素もある．前者はこれらのゲームでかなり基礎的なルール，後者は発展的または特殊なルールとみなすことができる．

3. 具体的な対象問題設定「戦術的 TBS」

内政要素のないターン制ストラテジーはタイトル間のルールのバラつきのため，AI 設計問題として統一的には扱いにくい．そのため本章で我々は内政要素のないターン制ストラテジーに共通する最低限のルール群を抽出して，本ジャンルでの AI 設計への入り口として取り組むにふさわしいゲームクラスの提供を狙う．そのゲームクラスを便宜的に「戦術的 TBS」と呼ぶ．また，後の章でゲームクラスの性質を例示する際に用いる TUBSTAP 環境の説明も本章で併せて行う．

3.1 具体的ルール

我々は前章で，ターン制ストラテジーを内政ルールの有無で区別する分類を述べた．内政要素のないタイプでは総じて従来型の木探索手法が適用できて，そのジャンル内で有効な AI 設計技術にはある程度の共通性があると考えられる．

とはいえ，AI 設計のための具体的な 1 つの対象問題として捉えるにしては，このジャンル内ではタイトル間に無視できないルールのばらつきも含まれる．例えば，新規に駒を増やす「生産」ルールの有無は工場マスの奪取の重要性をゲームに加え，AI 設計の指針に与える変化が大きい．

そこで我々はこのジャンルに取り組むために，その入り口となるべきゲームクラスを提供する事によってジャンル全体への広いアプローチを可能にしたい．他のアプローチとしては，具体的な 1 つの既存タイトルを対象問題に選ぶ事や，あるいは既存タイトルでのルール全てを包含するゲームクラスの提案が考えられる．しかし前者のアプローチについて言うと，表 1 の各既存タイトルはどれも他のいずれかのタイトルに含まれないルールを一個以上含むため，1 つのタイトルだけに特化した技術が探究されるリスクがあると考える．また後者のアプローチでは，ルールが複雑になりすぎて，解決が著しく困難になると予想する．

表 1 に挙げた既存ゲームのうち，内政の無いターン制ストラテジーに共通するルール要素を抜き出す事で，ジャンル全体で必須とされる部分問題のみの解決を試みる．具体的には，**｛複数着手性・駒の攻撃・HP・初期局面多様性・地形・相性｝**のみを備えたターン制ストラテジーを我々は提案する．**占領**ルールが含まれていないが，このルールがなくてもゲームクラスとしてはチェスに比べ大分複雑であるので，取り組みやすい複雑さのバランスを考えた結果である．現在の問題クラスにて技術の探究が十分に進めば**占領**や，（多くのタイトルに共通な）**生産**のルールも問題クラスに含めて扱うべきと考える．

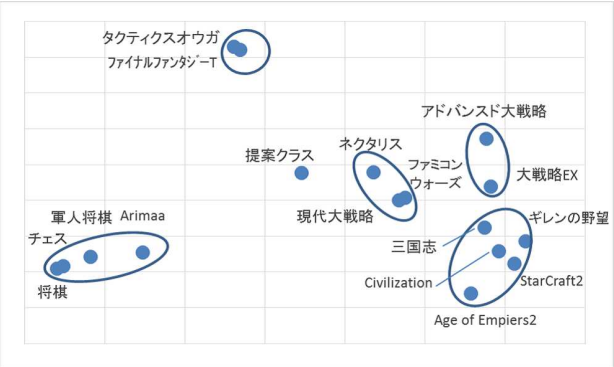

図 2 既存ゲームタイトルのルール要素によるクラスタ分析．ルール要素の違いがタイトル間の距離に反映される．同一座標にプロットされた 2 点は片方の位置を少しずらしてある．

Fig. 2 Cluster analysis of rules of existing turn based strategy games. The distance between two plots shows the difference of their rules. In case two plots would locate in the same coordinate, one of them was slightly shifted.

この**{複数着手性・駒の攻撃・HP・初期局面多様性・地形・相性}**のみのゲームクラスに我々は本稿中で「戦術的 TBS」という便宜的な名称を与える．この戦術的という語は，戦略シミュレーションの区分に由来する．プレイヤが指揮する戦場のスケールが広い順に「戦略的・作戦的・戦術的」と分ける区分が提唱 [22] されていて，内政のないターン制ストラテジーは戦場のスケールがおおむねその「戦術的」な区分に相当している．

このゲームクラスは，内政のないタイプのターン制ストラテジーを AI 作成を試みる研究者に，そのジャンルで（**占領**を除いた）最低限対処すべき問題を扱いやすい形で提供する事で，その解決を助ける役割を果たすと我々は考える．

3.2 ゲームルール群の俯瞰

提案ゲームと周辺ゲームとの関係を俯瞰する．我々はまず図 2 に示すようにゲームルール要素のクラスタ分析を行った．これは表 1 の各ルール要素の有無をバイナリ化した上で主成分分析を行って，第 1 主成分と第 2 主成分のスコアによって 2 次元平面にプロットしたものである．着手については（セミ RT を RT と同一とみなして）4 種のパターンがあるのでバイナリを 3 つ分（「複数着手制であるか」「RT 制であるか」「素早さ制であるか」）使った．主成分の導出は統計ソフト SMPP の ver. 2.20 により行われた．クラスタの分割は我々が恣意的に行った．ルール要素の違いが距離に反映され，近いゲームは類似したアプローチで対処できるとごく単純には見積もる事ができる．

この図にはまず左下にチェスや Arimaa のような古典的なボードゲームのクラスタがあり，それらに**{複数着手性・駒の攻撃・HP・初期局面多様性・地形・相性}**および**{占領・生産}**などを加えた基礎的なターン制ストラテジーが中央にまとまっている．戦術的 TBS（提案ゲームクラス）は，生産や占領がなく単純であるため，それらから少し古典的ボードゲームの側に寄っている．対して，右上には地形多層性や索敵や ZOC など複雑なルールが加わったゲームのクラスタがある．

上部には，キャラの成長要素があったり素早さパラメータをもとに駒の行動順が割り当てられる SRPG のクラスタがある．右下は内政またはリアルタイム性と索敵が加わったゲームのクラスタであって，これは内政のあるターン制ストラテジーや RTS から構成される．

こうして戦術的 TBS のクラスは，古典的ボードゲームと性質が近く，かついくつかの既存タイトルと共通なルール要素を備える．こうした性質により戦術的 TBS は，対象としての取り組み易さと，内政のないターン制ストラテジーが共通して含む多くのルール要素の解決を目指した問題の提供を両立できると考える．

表 2 TUBSTAP の駒間相性：各コマから各コマへの攻撃力

Table 2 The relationship of competitiveness between every two units in TUBSTAP. The value shows a coefficient to decide the combat damage.

		F	A	P	U	R	I
攻撃側	F（戦闘機）	55	65	0	0	0	0
	A（攻撃機）	0	0	105	105	85	115
	P （戦車）	0	0	55	70	75	75
	U （自走砲）	0	0	60	75	65	90
	R （対空戦車）	70	70	15	50	45	115
	I （歩兵）	0	0	5	10	3	55

3.3 TUBSTAP

我々は以降の章で，戦術的 TBS の局面例示に TUBSTAP プラットフォームを用いる．これは学術研究用に開発されたターン制ストラテジーゲーム用プラットフォームであり，戦術的 TBS の要素を満たす．生産および占領の要素は持たないが，反撃の要素を持つ．

TUBSTAP はシステムやルールの大部分が『ファミコンウォーズ DS2』をモデルとしている，2 人完全情報ゼロ和有限確定ゲームである．スクリーンショットを図 3 に示す．また主要なルールとして複数着手性や相性，駒の HP，隣接と遠隔の攻撃，移動，反撃，地形，などがある．勝利条件は敵の駒の全滅か，一定のターン経過時点で駒の HP 総和で一定値以上相手を上回ることとなっている．戦闘の条件や駒の初期配置は様々な設定から選ぶことができる．

使用可能な駒は 6 種類で，それぞれ攻撃時には表 2 に示す係数を用いてダメージが決定される．この係数は駒の相性の反映である．駒間で値が非対称的で，ある駒からみてダメージを与えやすい駒や，逆にまったく与えられない駒がある．またダメージは攻撃される駒がいるマスの地形にも影響される．

表 3 上段に示す係数によって，前述の攻撃用の係数と合

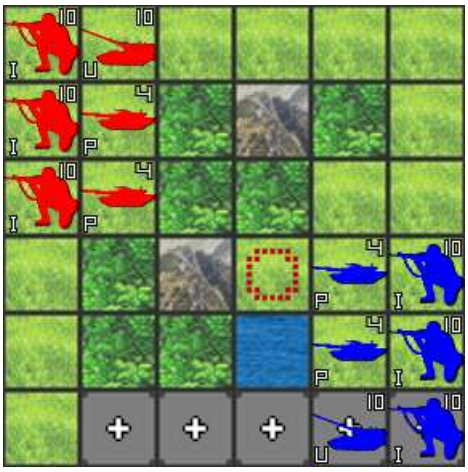

図 3 Tubstap スクリーンショット（ルール説明のために図 1 を再掲）．駒の色が陣営，数字が HP，文字が駒種類を表す．マスには草原や林，海などの地形が割り当てられている．例えば左から 4 列目のマスは上から順に，平原，山，森，陣地，海，道路の地形である．

Fig. 3 Screenshot of TUBSTAP. Color of unit shows its side. The number on each unit shows its HP value. The character on each unit shows its unit type. Each tile has its land type. T

表 3 TUBSTAP の地形が各駒に与える防御効果
Table 3 Land effect values in TUBSTAP

	平原	陣地	山	林	海	道路
F,A	0	0	0	0	0	0
P,U,R,I	1	4	4	3	0	0

表 4 TUBSTAP の地形が駒駒に課す必要移動コスト；駒の移動力は F, A, P, U, R, I でそれぞれ 9, 7, 6, 5, 6, 3
Table 4 Cost values for movement actions in TUBSTAP. The values for moving capacity are also assigned. That is, 9 for F, 7 for A, 6 for P, 5 for U, 6 for R, 3 for I

	平原	陣地	山	林	海	道路
F,A	1	1	1	1	1	1
P,U,R	1	1	∞	2	∞	1
I	1	1	2	1	∞	1

わせ，式 1 によって小数点以下切り捨てでダメージが計算される．攻撃後に相手の HP が 0 以下にならなければ，「反撃」といって，その相手からの攻撃が自動的に即座に行われる．

$$\frac{\text{攻撃力} \times \text{攻撃ユニットHP} + 70}{100 + (\text{地形効果} \times \text{防御ユニットHP})} \quad (1)$$

各駒の移動可能範囲は，移動力で決まる．F, A, P, U, R, I の移動力と地形に対する移動コストは表 4 に示される通りで，経路となるマスのコスト合計が移動力を超えない範囲内にある 1 マスに，各手番に移動ができる．味方の駒は通過できるが敵の駒は通過できない．移動し終わったマスの隣に敵の駒があれば U 以外の駒は攻撃ができる．攻撃してからの移動はできない．U は，自分から 2 マス~ 3 マスだけ離れている敵に一方的に攻撃を行える．

4. 主な既存 AI 手法

戦術的 TBS で AI 手法が受ける影響を論じるための準備として，古典的ボードゲームでよくみられる AI 手法や，TUBSTAP プラットフォームで実装例が見られる手法について本章で説明する．

4.1 Min-Max 型木探索と状態評価関数

ゲーム木を Min-Max 探索し，読み深さの限界に達したところでノードの好ましさを状態評価関数で数値化する手法である．Min-Max 探索の代わりに高速な $\alpha\ \beta$ 探索を一般に用いるが，他にも枝刈りによる高速化がよく行われる．また状態評価関数は，人手による設計や，機械学習の自動的な調整による設計 [23] がある．

ただし本稿の以下の章では，戦術的 TBS の初歩的な状態評価関数を例に用いる場合，その形式を単純な HP と駒価値の線形和で与える．すなわち，局面にある味方の駒それぞれの，{(駒種類に与えられた静的な定数)×(駒 HP)} の総和から，敵駒それぞれについての {(駒種類に与えられた静的な定数)×(駒 HP)} をマイナスした値を，その局面に対する評価値とする．

4.2 行動評価関数による単一着手選択

ゲームにおける行動に評価関数を用いることで，現在のプレイヤにとって最も好ましい次行動を選択するような手法である．例えば移動行動には移動先のマスの地形効果の量を，攻撃行動ならば与えるダメージと反撃で受けるダメージの差を評価値として与える評価関数が考えられる．

1 ユニット行動分だけの深さの Min-Max 型探索手法を，現在局面の評価値でバイアスされた局面評価関数とともに使っているような，4.1 節の手法の一種であるという見方もできる．

4.3 MCTS

モンテカルロ木探索（MCTS）は，モンテカルロシミュレーションによって局面の良さを評価しながら，段階的にノードを展開して探索を深めていく手法である．ノードを展開しない場合には原始モンテカルロ法となる．次項で述べる UCT 探索が一般的である．また TUBSTAP プラットフォームにおいては，UCT 探索と異なる MCTS または原始モンテカルロ法の実装例もあるのでこれも続けて述べる．

4.3.1 UCT

Upper Confidence Bounds applied to Trees[24] は，ノードのシミュレーション勝率の不確かさを考慮して，探索数が少ないノードには探索をうながすボーナスを与える MCTS である．こうして計算される，ノードに探索の優先度を与える数値を UCB 値という [25]．その UCB 値はパ

ラメータの係数 C を用いて以下の式 2 で計算される．

$$(\text{平均勝率}) + C\sqrt{\frac{ln(\text{親ノード訪問回数})}{\text{訪問回数}}} \quad (2)$$

そして一定回数のシミュレーションを実行されたノードは展開によって，その子ノード群をシミュレーションの開始ノードにして自分をシミュレーション評価の対象ではなくし，子ノードのシミュレーションに応じた評価値の更新を行うようにする．

4.3.2 PW を伴う UCT

Progressive Widening（本稿では以降 PW と略記する）は，探索候補となるノードにフィルタをかけ，探索が進むにつれて徐々に読む着手の種類を増やしていく手法である．探索の最初期に根ノードの子ノードうち，例えば（何らかの指標で）上位 a 個のみを訪問の対象とする．そして全ノードの訪問数が増えるにしたがって，訪問の対象となる子ノードの個数を b 個，c 個と順に増やしていく $(a < b < c)$．

このような工夫をほどこす事で，ある種の有望そうな着手を深く重点的に調べてから，徐々にそれ以外の手を読みを広げていくような探索ができる．性質としては UCT 探索に比べて，ノード評価値に大きな影響与えるような特定の手筋を読み取りやすい．

4.3.3 深さ限定型モンテカルロ（DLMC）

モンテカルロ法のランダムシミュレーションを一定の深さで打ち切って局面評価関数を適用し，その評価値をシミュレーション報酬に代える方法が提唱されている [26]．

TUBSTAP 環境では，その手法を原始モンテカルロ法に加えて用いた実装が報告されている [13]．この手法ではまず 1 手番における合法手組み合わせをいくつかランダムにサンプリングし，それらを一定深さだけシミュレーションする．一定深さ到達後の局面を状態評価関数で評価し，シミュレーション回数の平均値がノード（1 手番のある合法手組み合わせ 1 組）の評価値となる．浅い深さのノードに短いシミュレーションを多く適用するので，最序盤など局面のなんとなくの良さを見積もる場面で効果を発揮すると考えられる．

4.3.4 攻撃行動探索（AAS）

攻撃行動を中心に読む手法である [13]．手番の最初，自軍に N 個の駒があるとして，ある L 個 $(L \leq N)$ からなる攻撃行動の組合せ（正確には順序を考慮するため順列）それぞれに，残り $(N - L)$ 個の駒の行動（ルールベース等により決定）を付けたして，その手番の行動組み合わせを生成する．そしてそのような 1 手番中の自軍の行動組み合わせそれぞれの結果を読み，局面評価して，その手番に行うべき着手組合せを決定する探索法である．ある限定的な攻撃手順の組み合わせを見つけ出すことに適している．

5. 考慮すべき要素 1 : 可能着手の組合せ爆発

本章から戦術的 TBS が AI 設計問題として持つ特徴について記述する．戦術的 TBS がゲームとして持つ性質のいくつかは AI の設計や動作に大きな影響を与える．我々はそうした性質を，AI 設計時に考慮すべき要素として 3 つ述べる．まず本章は読み手番数に対する可能着手の組合せ爆発について述べる．

5.1 概要

戦術的 TBS には一回の手番で複数の駒を好きな順序で動かせる複数着手性がある．そのルールのために手番あたりの合法手の数は組み合わせで非常に大きな数となる事が，ターン制ストラテジーの一ジャンルの Turn-Based War Chess game を題材に Nan らにより指摘されている [29]．1 駒あたりの行動数を n, 駒数を r とすると 1 手番の行動の組み合わせは $(n^r \times r!)$ 通りにもなる．

例えばそれぞれ 10 通りの行動が可能な駒が 6 つある状況を考える．この程度の数の規模は戦術的 TBS では珍しくないが，上述の計算により，1 手番に可能な行動組み合わせは 7 億 2000 万通りにもなる．他のゲームでの 1 手番ごとの平均合法手数が，チェスで 35，囲碁で 250 であることを考えても非常に大きな値である．

5.2 AI 手法への影響

手番ごとの合法手組合せの膨大さのため，チェスや将棋のように Min-Max 型探索を全幅探索するアプローチは困難を伴う．例えば，ある先鋭の将棋 AI の読み速度は秒間 10 万局面前後と言われるが [27]，前項で述べた条件下での 7 億 2000 通りの合法手組合せを全て読もうとすれば，1 手番の結果を調べるだけでも 7000 秒かかる．ちなみに今回我々が用いるプログラムでは読み速度は秒間数万局面なので，より多くの時間が必要になる．そのため駒数が少なくない局面では，何かしらの対策をほどこさなければならない．以下，想定できるアプローチを個別に論じる．

(a) 局面の分離：まず人間プレイヤが直感的に行っているアプローチとして，考慮する局面の分離がある．全体の大きな局面のうち，注目すべき小さな局面に切り分けて，それぞれを十分に深く読む．例えば前述の 10 通りの行動が可能な駒 6 つずつの局面を，2 駒対 2 駒の小局面 3 つに分離して独立に扱えて，各小局面ごとの最善手さえ探せばいいものとする．そのとき，1 手番で読む局面は $10^2 \times 2!$通り $\times 3$ 小局面 で，600 通りのみ読めば良い．この減少の度合いは劇的であるが，とはいえあくまで局面の分離が確かに可能で，その分離可能性を AI が正確に判断できるときのみ成立するアプローチである．

(b) 着手の絞り込みと深い探索： 有望な着手に読みを絞る試みは，人間プレイヤが行うのみならず，他のゲームでAI が木探索を行う際にもよくみられる．$\alpha\,\beta$ 探索もその一例で，根ノードの Min-Max 評価値の計算結果を変えないことが保障されるため広く用いられる．また各ノードごとにあきらかな悪手を探索から除くことも，根ノードの Min-Max 値を変えてしまう恐れがあるものの，よく行われている．ただし前項で例示した条件下で，いわゆる『3手の読み』（自分の手，相手の手，そのあとの自分の手までの読み）を１０秒以内に行おうと考えれば，候補手を 7 億の中から 100 手 [*1] に絞らなければならない．ちなみに $\alpha\,\beta$ 探索で枝刈が最善に行われる場合でも，約 1000 手に絞り込む必要がある [*2]．つまり AI 設計者には，局面ごとの有望そうな着手に大きな絞り込みをかけるための工夫が求められる．

例えば駒の行動順序を，固定された 1 通りのもの（HP の高い順や，完全にランダムな順）のみ考慮して行動を生成すると手番あたりの枝数が，$\frac{1}{\{\text{手番プレイヤの駒数}\}!}$ 倍になる．または，ターン制ストラテジーは概して駒の移動行動の数が攻撃行動（移動してからの攻撃も含む）の数より大きくなりがちなので，各ノードで移動行動全てを探索から除く枝刈りを行えば，探索量がかなり大きく削減される事が期待される．

(c) 浅い探索と精度の良い評価関数：上記 (b) の方法とは逆に，探索深さは浅くとも評価関数を洗練するアプローチも考えられる．浅い探索とは例えば，複数の駒のうち 1 駒分の行動の結果を評価するような木探索である．前項の条件でも読みの範囲を「1 手番すべての行動の結果」から「1 駒の行動の結果」に絞れば想定するパターンは 10×6（=60）通り [*3] で済む．そしてそうした浅い探索を行う場合には，探索全体の性能は状態（または行動）評価関数の精度に大きく影響されると考えられる．

(d) UCT 探索：UCT 探索も有望そうなアプローチである．まず UCT 探索は 1 駒の行動を 1 ノードに対応させた場合は，最初の探索は深さが浅い．そして，シミュレーションの結果により有望そうな着手に探索を集中させながら探索を深くしていく挙動を見せる．よって，UCT 探索は上記 2 つの「(b) 着手の絞り込み」（ノードの探索を後回しにするのか打ち切るのかの点で違いはあるが）と「(c) 浅い探索」を折衷して行うアプローチとも言え，戦術的 TBS 環境下でもある程度の効果が期待できる．ただし後述のように駒の防御的な陣形配置を行う必要がある場合は，探索を深いノードに掘り下げていくより多様な浅いノードに探索資源を集中させた方が良いような場合もある．

[*1] $100^3 = 10$ 万局面 × 10 秒
[*2] Knuth ら [30] の計算に基づく
[*3] もちろん 1 駒の行動しか決まらないので，1 手番を終えるまでには $10\times6+10\times5+10\times4+10\times3+10\times2+10\times1=210$ 通りの読みは必要になる．

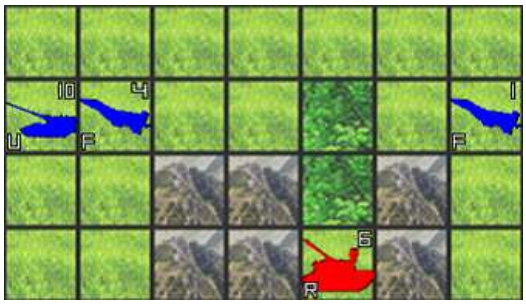

図 4 赤はナイーブな浅い読みで失敗する場面．HP が高い左の F を攻撃しに行くと，次手番で敵の U に駒が破壊されてしまう．

Fig. 4 The position where a shallow and naiive tree search cannot find winning moves of red player. If the unit R attacks the F unit with higher HP than the other F, the R should be destroyed by U in the following turn.

5.3 影響の例

手番ごとの着手の多さから AI 手法が受ける影響の例を，図 4 の局面の赤のプレイヤを例にして述べる．赤の手番であるとする（以降，本稿で扱う局面図は全て赤の手番で始まるとする）．これを，ある大きな局面の部分局面だと仮定し，ナイーブには十分な深さの探索が行えないものとする．つまり，他の部分局面には味方の駒が多くあって，手番の全ての行動の組合せを読もうとすると枝数が膨大になり，深さ 1 を超える全幅探索は適用できない状況であると仮定する．

この局面は赤にとって深さ 1 の全幅探索と単純な局面評価関数（4.1 項参照）を用いても適切な行動を導出できないようになっている．赤の R は青のもつ 2 つの I のどちらかを最初に攻撃して除去できる．左の F の方が HP は高いため，赤は右の F より左の F を攻撃すると HP 線形和による局面評価値が高くなる．そのため，深さ 1 の木探索と上述の単純な局面評価関数を用いた AI は左の F への攻撃を選ぶ．

しかしその場合は次に相手の U に R を攻撃されて赤は即座に負ける局面である．一方で右の F を攻撃すれば赤は次の手番以降に U の遠距離攻撃の圏外（U の周囲 1 マス）から一方的に U を攻撃して戦闘に勝つ．つまり赤にとっては何らかの適切な探索の工夫が必要な局面である．以下，順に前項のアプローチを適用した場合の影響を考察する．

(a) 局面の分離:この局面を完全に独立したものとして全体から分離すると，駒の可能行動を 10 程度と見積もって深さ 2 までの全幅探索で，葉局面の数は $1!\cdot10^1\cdot3!\cdot10^3$ でせいぜい 10^5 に達しない程度である．これは他局面の探索に対し，単純に加算される因子となるので，十分に現実的な

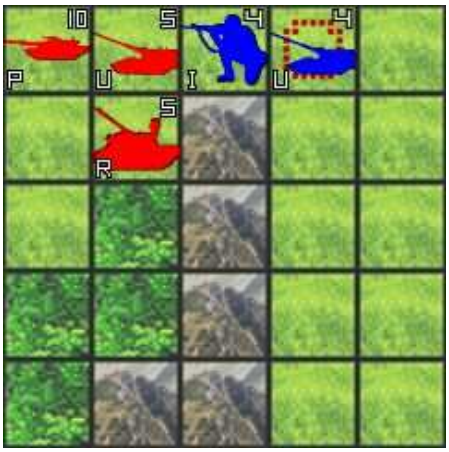

図 5 移動行動を読みから外すと失敗する局面の例．赤は U を移動させないと敵の I を除去できず，攻撃が止まってしまう．

Fig. 5 A position where a movement action is needed to win for red player. Red player must move his U at first.

探索量である．この局面で深さ 2 まで読めれば，左の I に攻撃した後に青の U に攻撃されて負ける枝が読めるので，負けの行動が回避できる．

(b) 着手の絞り込みと深い探索：攻撃を伴わない移動行動の枝刈りと，行動順序の固定により，探索を相手の手番の行動後（深さ 2）まで深くすると，左の F への攻撃行動の選択を同様に回避できる．このときの評価局面数は，局面全体の合計で赤と青が 3 通りの攻撃行動選択肢を持つ駒を各 6 つ所持しているとすると，$3^6 \cdot 3^6$ で 10^6 より低い程度である．ただしこのとき駒の行動順序組合せも考慮しようとすると，$6! \cdot 3^6 \cdot 6! \cdot 3^6$ で 10^{11} 程度にも局面数が膨れ上がってしまう．またこうした枝刈は，図 5 のように移動行動が正解となる局面で適切な着手を見落とすリスクも持つ．この局面では手番の赤は U を移動して味方に道を開かないと最善行動がとれない．

(c) 浅い探索と精度の良い評価関数：局面評価関数に改良をほどこす．例えば，こちらの行動終了した駒が相手の残存した駒から次手番に受けうるダメージを評価関数への十分大きな負のバイアスとして適用すると，赤は深さ 1 の探索であっても，初手で左の F より右の F への攻撃を選ぶ．

とはいえ，評価関数のオフラインの洗練は 6 章に述べる理由より難しく，オンラインなアプローチによる洗練もコストがかさむ．例えば先ほどの「残存した敵駒から受けうるダメージ」の計算には，敵駒と味方の駒を結ぶ経路のうち，敵駒の移動コスト以内で通過できるものがあるか調べなければならない．

(d) UCT 探索: UCT 探索は，この局面で右の F に攻撃するノードと左の F に攻撃するノードを訪問する．そして後者のノードをさらに展開すると子ノードには赤の R が青の U に攻撃され即座に負けるノードが現れる．そのため深さ 2 の全幅探索を行うには不十分な計算資源しかない場合，適切な枝のみを確率的に選んで UCT 探索は負けを回避しうると考える．

我々は，UCT 探索がそうした計算資源の制限下でも確率的に正解を導き出す事を示すための実験を行った．図 4 の局面を付録 A.2.1 に示す UCT 探索 AI に解かせる．

図 4 の局面を Min-Max 探索と HP 線形和の評価関数で解かせる場合，その思考時間は，深さ 1 なら 1.0 ミリ秒以下，深さ 2 なら 503 ミリ秒である．手番あたりのシミュレーション数を 100, 1000, 2000 としたとき，思考時間は各 20.0, 226, 388 ミリ秒かかり，それぞれ 100 回の試行中 13, 69, 100 回正解を導いた．つまり，この局面で UCT 探索は深さ 2 の全幅探索よりは少ない計算資源で確率的に正解を導き出す事が示された．

6. 考慮すべき要素 2 ：局面の形勢判断に求められる即興性

次いで説明するのは，戦術的 TBS の局面の形勢判断が要する即興性についてである．このゲームの局面の形勢判断は，事前の知識に多くを頼って行うことが難しく，その場ごとの読みや感覚に大きな部分を頼ることになる．これはゲームのルールのうち，初期局面の任意性，マスの地形，駒の相性の 3 つに根ざした性質である．

6.1 概要

戦術的 TBS では作者やプレイヤがデザインした初期局面から対戦が始まり，普通は様々に初期局面を変えながらゲームを遊ぶ．初期の駒の数と種類や配置，盤のマスの地形の配置，盤のサイズまでがゲーム毎に変化して，小規模なものなら 4 マス四方程度の盤に数体の駒，大規模な場合では盤の一辺のマス数や盤上の駒数がともに数十に及ぶこともある．しかもそれらは両プレイヤにとって非対称的である場合も多い．こうした初期局面の多様性は，チェスや将棋と対照的である．

定跡利用の困難：チェスや将棋のボードゲームでは常に同じ局面からゲームが始まるため，出現頻度の高い序盤の局面に知識が蓄積される．のちのち不利になりがちな局面はプレイヤに避けられるようになっていき，いわゆる序盤定跡がゲームに形成されていく．一方で戦術的 TBS は毎回違った局面からゲームが始まるためそうした序盤定跡はできにくく，最善な行動を常に新しく考える必要がある．

戦術的 TBS にも一応は序盤定跡に類する行動指針はあると考えられる．例えば「耐久力の弱い駒は味方の駒で囲んで攻撃される機会を減らす」といった行動指針がそれにあたるが，これらは大まかで，また多少曖昧な指針である．チェスや将棋のような具体的局面への研究の蓄積ではなく，序盤定跡ほどの高い信頼性は期待しづらい．

多様な地形マス配置の利用：また盤上の地形マスの様々な初期配置もプレイヤの行動判断，特に駒の陣形に関して，即興的な思考を多くうながす．戦術的 TBS の地形は，その上にいる駒の HP を減りにくくする事がある．そのためプレイヤは有利な地形マスに駒を置きながらも駒間の位置関係をなるべく有利なものにしたい．そして盤上の地形マスの配置は前述のとおりゲームごとに変化するので，毎回そうしたバランスを新しく考える事が重要になる．

相性により複雑に変動する駒価値の扱い：さらに戦術的 TBS の駒の相性のルールは，自軍の駒の価値に関してもその場での思考を多く促す事がある．チェスや将棋ならば駒の価値は比較的安定している．クイーンや竜王のような強力な駒は，駒同士の位置関係によって一時的にうまく働かないことはあっても，また位置関係が変わればだいたい局面に強く有利さをもたらすことが期待される．しかし戦術的 TBS では位置関係によらずとも，ある駒の効力が一切封じられることがある．

例えば TBSTAP で P は A に攻撃ができずに一方的に攻撃を受けるため，A だけの大群を相手にする局面で P はほぼ常に無価値である．さらにこうした駒の有利不利の関係は，三すくみや四すくみの循環的構造になっていることが多いため，P が高い価値を持つ一方で A が限りなく低い価値しかもたない局面も，互いの陣営の駒編成によっては現れる．そのためプレイヤは相手のプレイヤの持つ駒の種類に注意しながら自群の駒の価値への判断をその都度更新し続けていく必要に迫られる．

6.2 AI 手法への影響

これらゲームの性質は AI の局面評価（または行動評価）に影響を与える．局面評価に際して，主にオンラインなアプローチの有効性を増すと我々は考えている．まず初期局面の多様さから，定石に頼っての序盤局面の評価は困難になる．加えて，局面評価をオフラインな機械学習手法に任せることも不可能と言えないまでもかなり難しくなると考える．なぜなら，ある駒価値の割り当てはほぼ同じ駒編成の局面間でしか共有できないためである．そして局面の多様性により十分な汎化性能をもつ学習用データの調達が難しくなる．

さらにボードゲームの局面評価には往々にして駒同士の位置関係が情報に用いられるが，戦術的 TBS での高精度な局面評価のためには，駒がいる地形マスの並びも情報として扱う必要があると考える．そして地形マスの配置はゲームごとに異なるため，様々なパターンを形作るのでそれぞれの初期局面が与えられるたびにその地形マスの配置に対応していく必要がある．

加えて，残存する駒の価値もボードゲームの局面評価によく用いられる．チェスや将棋の AI の局面評価関数にも，駒の価値を反映させる項が含まれる．それらは静的な数値であり，駒の位置関係の項とは独立であることが多い．しかし戦術的 TBS では駒の，位置関係とは独立したそれ自体の価値が局面次第で大きく変わる．よって，高精度な局面評価のためには状況ごとに動的に駒の価値を見積もっていく必要があると考える．

こうして局面評価にオンラインなアプローチの有効性が期待される環境下で，最も単純に有効性が見込めるのはモンテカルロ木探索ベースの手法である．なぜなら局面評価がゲームのシステムにそったシミュレーションを基に行われるためである．とはいえ Min-Max 型の探索に局面評価関数を併せた手法であっても，計算時間を使って適宜有用な陣形や駒価値の調整を行うことで効果的な動作は期待できる．オフラインのアプローチの場合は，特定のゲームでのみ通用する学習を行う事は可能である．

6.3 影響の例

駒価値の変動とそれが AI 手法の有効性に与える影響を具体例で示す．図 7 および図 9 の局面を想定する．これらの局面は先手である赤が有利で，読みの中で着手後の局面を正しく評価できれば勝利できるように作られている．

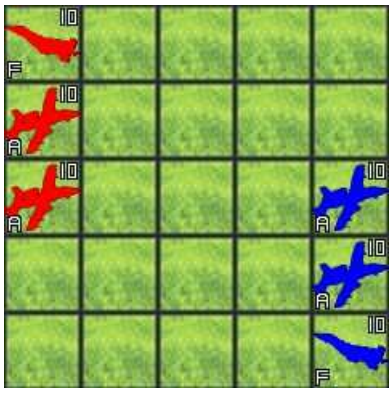

図 6 赤は F を A より重く扱うことで勝利できる局面．

図 7 A position where assigning larger value to F than A results in win.

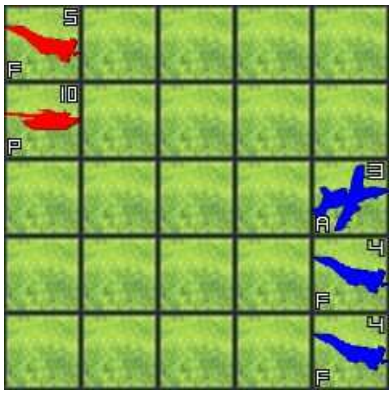

図 8 赤は，図 7 と逆に，A を F より重く扱うことで勝利する局面．

図 9 A position where assigning larger value to A than F results in win.

表 5 異なる駒価値割り当てによる対 UCTAI100 戦実行時の勝率
Table 5 Win rate againt UCT AI over 100 matches with different ways of unit evaluation

赤 AI	図 7 局面勝率 (%)	図 9 局面
A 偏重	0	100
F 偏重	100	0
UCT	97	28

それぞれの場面で，4.1 項に述べた，駒の種類と HP に基づく局面評価モデルの使用を想定する．図 7 は，A よりも F に高い価値を割り当てることによって適切な行動を導ける．この局面で，F は F にも A にも攻撃できるが，A は F にも A にも攻撃ができない．そのため相手の F さえ取り除けばこちらの駒はもう攻撃されなくなり，確実に勝利する．

一方で図 9 の局面は F より A に高い価値を割り当てないと適切な行動を選べない．先手の赤は，F で相手の F を攻撃するか A を攻撃するかの選択肢を持つ．ここで A を標的にすれば，青に残った F は赤の P に攻撃ができないため，最悪でも制限ターン超過の HP 判定によって赤の勝ちが約束される．一方で F を標的にした場合は，次の手番で相手の F2 体に赤の F が破壊され，赤は P のみで青の A と F に立ち向かわねばならない．P は A に攻撃できないが，A は P に一方的に攻撃できるため，これは青の勝ちである．

我々は，駒価値の異なる局面に対する局面評価法のふるまいを比較するため，静的な評価関数の Min-Max 木探索プレイヤとシミュレーションにより局面を評価する UCT プレイヤを対戦させた．赤のプレイヤとして以下の 3 種を用いて対戦実験を行った．

(1) **A 偏重型 Min-Max**：(4.1 項のモデルで) A の駒価値を 100，F の駒価値を 1 とした局面評価関数を用いる，1 手番分の深さの Min-Max 木探索プレイヤ
(2) **F 偏重型 Min-Max**：F の駒価値を 100，A の価値を 1 とする局面評価関数を用いる，1 手番分の深さの Min-Max 木探索プレイヤ
(3) **UCT 探索**：UCT 探索により着手を決めるプレイヤ．手番あたり 1000 シミュレーション行う，付録 A.2.1 の仕様のもの

青のプレイヤは常に UCT 探索 AI を用いるが，これも仕様は付録 A.2.1 に等しいシミュレーション回数は手番ごとに 6000 回とした．それぞれの局面は 16 手番の経過により，陣営の駒の HP 総量が大きいプレイヤが判定勝ちとなる．

さて，それぞれの試合 100 戦の結果を表 5 に示す．F より A を重視すべき局面では，A を重視した評価関数を用いた AI が全勝し，A より F を重視すべき局面ではその逆である．UCT 探索は勝敗が確率的である．

このように戦術的 TBS での局面評価は，重視するべき駒の種類が互いの駒の状況により変わる．UCT 探索のシミュレーションベースの局面評価でも 2 つの局面で確率的に正しく局面を評価できたが，シミュレーションは HP 線形和の関数による局面評価より計算時間の消費は大きい．一方で，オフラインな局面評価関数は初期局面に対応した適切な局面評価関数である場合なら，高速に正しい評価を提供できると考えられる．

7. 考慮すべき要素３：攻撃がもたらす影響の潜在的大きさ

３つ目の要素として取り上げるのは，1 手番の攻撃行動組み合わせが局面にもたらす影響の大きさについてである．

7.1 概要

戦術的 TBS では複数着手性があるため，大勢の駒が一斉に相手に攻撃すれば形勢が大きく変化する事が多い．例えば互いの駒が 30 ずつあっても，有利な相性を持って全部の駒が敵の駒にそれぞれ攻撃して除去に成功した場合，片方のプレイヤの駒数はたった 1 手番で 0 になりうる．これは将棋等で言えば，王将の周囲で陣形を組み護衛していた駒全てが，たった 1 手番で敵により全て取り除かれるに等しく，盤上に与える変化は劇的である．

ただしそうした 1 手番の攻撃行動の影響力を十分に引きだすためには，それに特化したある種の思考が必要になる．駒ごとの適切な攻撃対象の選択は当然として，それに加えて「攻撃を行う位置」や「攻撃を行う駒の順番」なども綿密に考慮した先読みが求められる場面も多い．攻撃を行う位置が不適切だと，自軍の他の駒の移動や攻撃の妨げになる場合がある．また攻撃の順番に関しては，特に敵側の重要な駒が他の頑丈な駒に囲まれている場合，こちらの駒の攻撃順序次第でその重要な駒の除去の成否が分かれることがよくある．重要な駒というのは，例えば攻撃力が著しく高い一方で相手からの攻撃には弱い駒の事で，こうした駒を除去または適度に損傷させる事は戦局に与える影響が大きい．

さらにこうした攻撃行動組み合わせの影響力の十分な引き出しは，自軍の「防御」にもつながる点で重要である．というのも，相手の手番に十分に敵の戦力を削げなければ，次の相手の番にこちらの駒が敵の一斉攻撃にさらされる．往々にして攻撃を仕掛けたプレイヤの駒はバラバラに陣形が乱れがちで，相手にとっては最も効果的な攻撃行動組み合わせが容易に発見できる局面を作り出してしまう．

そしてそのような攻撃行動にまつわる性質は，守備的な陣形構築の重要性にもつながってくる．打たれ弱い駒を他の駒で守ったり，相手の駒それぞれの攻撃射程を考慮して自軍の駒を配置するなどの工夫で，相手の総攻撃がこちらに与え得る被害の規模は大きく変わる．そうして相手の攻

表 6 攻撃行動組合せ探索が有効な局面での最善手発見確率：100 回

Table 6 The success ratio of finding the best move over 100 times in a position where the combination of attack actions have large effect.

赤 AI	成功率 (%)	平均思考時間（ms）
UCT	17	1,720
PW 付き UCT	89	1,500
AAS	100	176
DLMC	0	1,779
Min-Max	100	4,895

撃による被害を適切に抑えることに成功すれば，前述のように，相手の乱れた陣形にこちらが総攻撃を加えて一気に勝ちきれる展開が予想される．そのような事情のため，敵駒の軍勢との接近時に適切な守備性を備えた陣形を構築する能力はゲームの勝敗に大きな貢献を与える．

7.2 AI 手法への影響

AI がしかるべきタイミングにおいて攻撃行動組み合わせの精密な読みを怠った場合，適切な総攻撃による潜在的な利益を逃すだけでなく，続く手番で相手側からの総攻撃により大きな被害を受ける展開が予想される．その精密な読みのためには，どの駒がどの駒に攻撃を加えるかという選択肢だけでなく，どの駒が「どういう順番で」，「どのマスから」相手に攻撃するかも省略せず，可能な行動を網羅的に探索する必要があると考えられる．

そうした網羅的な読みを提供するための直接的な手段には Min-Max 型の木探索がある．ただしその際には 5 章で述べたような行動組合せの爆発を考慮に入れる必要もある．

また，守備的な陣形を適切に構築する読みも AI にとって重要になる．この能力が稚拙な場合は，AI は相手プレイヤに無防備な陣形で近づいてすぐ負けてしまうか，あるいは相手プレイヤに自分からは近づけない戦いのスタイルになると予想できる．

どのような読みを行えばこうした適切な守備陣形に駒を配置できるかは，直感的にはあきらかではない．ただし既存の研究によれば，4.3 項で述べた DLMC 手法による探索が，このような守備的な陣形を作る読みに適していると報告されている [13]．

7.3 影響の例（攻撃）

我々は，まず攻め手筋の精密さに応じて戦果が大きく変わる状況を図 10 に示す．手番側の赤は先手をとって攻撃ができ，効果的な攻撃行動の選択により勝てるよう設計された局面である．しかし，相手の陣形には攻撃を仕掛けられるポイントが狭く，少しでも行動の選択を間違えると自軍の行動済みの駒が邪魔になってしまって残りの駒が攻撃できなくなってしまう．

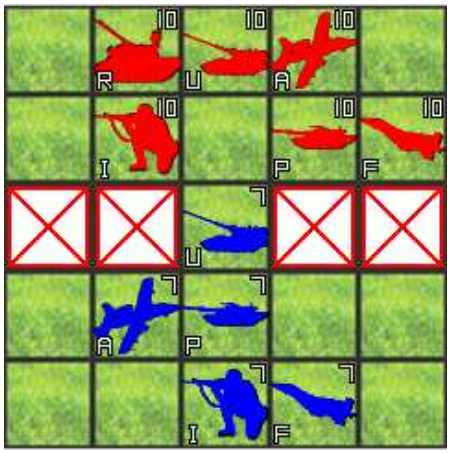

図 10 攻撃重視の探索が重要になる局面：赤は敵陣への入り口が狭く，また行動後の味方が邪魔にならないように，行動順序と攻撃対象を注意深く調べなければならない

Fig. 10 A position where search that focuses on attack actions works effectively. Red player must decide carefully the order in which his units make actions and the target units of attack actions.

この局面で，赤側の AI として 4 章で述べた，PW 付き UCT 探索，UCT 探索，攻撃行動探索（AAS），深さ 1 手番分の Min-Max 探索および DLMC を用いて行動させ，最適な手順をどれほどの思考時間で発見できたかを調べる．それにより，1 手番の適切な攻撃行動組み合わせ発見に対する各手法の得手不得手を明らかにする．PW 付きの UCT 探索は一般に，通常の UCT 探索に比べて探索資源を特定の手順に絞り込んで探索を進めるために，このような多くの行動組み合わせから特定の手筋を発見することに向くと考えられる．この局面では赤が適切な 1 手番の行動で青側を全滅できるので，その手筋を最善手とし，最善手発見の成功率を観察する．各 AI の設定は付録 A.2 に示される通りである．ただし PW つき UCT と UCT は，最適手順の発見を目的とするためにシミュレーションを 1 手番の深さで打ち切って青側に駒が残っていれば赤の負けとみなした．手番ごとのシミュレーション回数は 20000 回である．PW つき UCT も同様の仕様をベースにするが，AAS の読む攻撃行動組合せ長さパラメータは 6 に設定し（$L = N$），局面評価関数は重みの等しい HP 線形和である．Min-Max 探索も同様の局面評価関数を用いる．DLMC は 200 個サンプルした 1 手番行動組み合わせのセットに，1 手番深さのシミュレーションを 100 回ずつ適用する．試行は各 100 回である．このときの正答率と平均思考時間は表 6 のようになった．

まず AAS は決定的な挙動なため，攻撃着手の組合せを発見して確実に正しい行動を導けた．攻撃が可能な限りは攻撃行動しか調べないため探査時間も短い．深さ 1 手番分の Min-Max 探索は正しい着手組み合わせを発見したが，計算時間は AAS より多く費やしている．DLMC は，手番の全

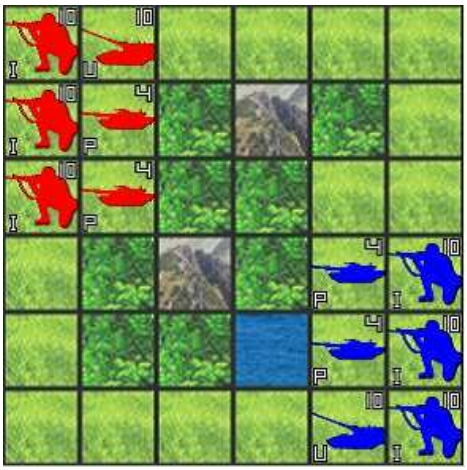

図 11 守備陣形の構築が重視される初期局面 A

Fig. 11 A position A where defensive formations have large effect.

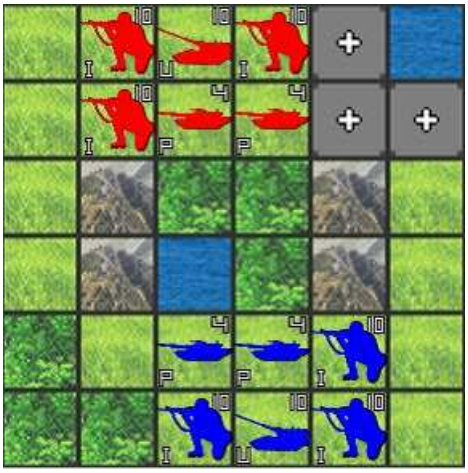

図 12 守備陣形の構築が重視される序盤局面 B

Fig. 12 A position B where defensive formations have large effect.

行動組み合わせから数通りの正解をサンプリングする事で正解は原理的に可能だが，今回の試行では 1 回も正解しなかった．そして UCT は PW を併せて使うことにより，効果的な攻撃行動組み合わせの発見に成功し勝率が伸びたと考えられる．対する通常の UCT 探索は多くの場合に効果的な攻撃手筋を発見できず，相手の反撃によって負けが勝ちを上回ったと考えられる．このように戦術的 TBS では特定の局面で攻めに特化した探索の工夫を加えることで，特定の最善手発見の見込みが大きく変動してしまうことがある．

7.4 影響の例（防御）

次に，守りに適した陣形の構築の効果性を図 11 から 12 の局面の対戦実験により例示する．これら各局面で序盤の最善な戦略の具体的手順などを論じることは難しい．しかし全体的な特徴として，互いにすぐに攻撃は届かないが，中央付近に有利な地形効果をもたらすマスも点在している．

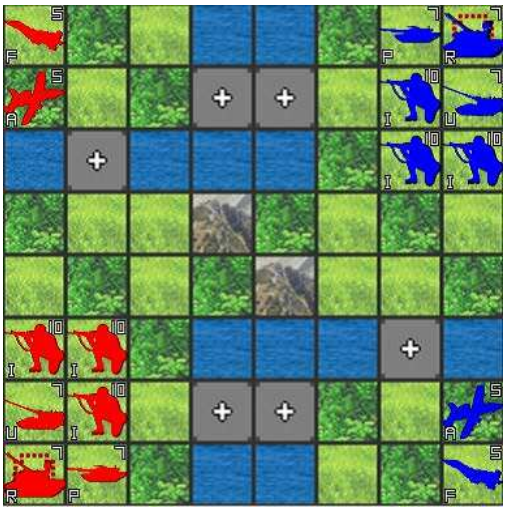

図 13 守備陣形の構築が重視される序盤局面 C

Fig. 13 A position C where defensive formations have large effect.

表 7 守備的な陣形構築の精度が勝敗に影響する状況の戦闘: 対 PW つき UCT 探索 AI 時 1000 戦勝率．カッコ内は 95%信頼区間

Table 7 Win rate values on positions in which forming defensive formation is important, against UCT with PW over 1000 matches. The values between blankets show the 95% confidence intervals.

	局面 A	B	C
	DLMC 先手時		
DLMC	48.6%(±6.2%)	55.5(±6.2)	73.7(±5.5)
	DLMC 後手時		
DLMC	51.4(±6.2)	60.0(±6.1)	62.4(±6.1)
	合計		
DLMC	50.0(±4.4)	57.8(±4.3)	68.1(±4.1)

さらにところどころ「山」のマスが障害物となって駒の移動を部分的にさまたげる地勢上の構造もみられる．よって障害物や地勢を利用して相手に攻撃の機会をあまり与えないように有利な地形マスに駒を布陣できれば，戦いが有利になる局面である．

我々はこの局面を使い，序盤に有効な陣形を形成する能力の手法による変化を観察するため DLMC と PW つき UCT の性能を比較した．この両者の対戦によって，シミュレーションの深さと木探索のノード深さを浅くして様々な着手に多くのシミュレーション回数を割り振る探索と，狭い手筋の深い読みのどちらが序盤の有効な陣形形勢に貢献するのかを調べる事を狙う．

我々は図 11 から 13 の局面で赤プレイヤに DLMC，青プレイヤに PW つき UCT を割り当てて，500 戦，赤プレイヤに PW つき UCT，青プレイヤに DLMC を割り当てて，500 戦の計 1000 戦ずつの対戦を行った．ただし，それぞれの守備的な陣形の構築の精度をより明確に比較する目的で，DLMC の 2 手番目以降は PW つき UCT が操作する．

このPWつきUCTはDLMCの対戦相手と同じ設計とパラメータを持ち，つまり，最初の手番（主に布陣に関わる行動からなると考えられる）だけAIのプレイヤの探索手法が違う．これにより最初の陣形の構築の精度が戦果に反映されやすくなると考える．PWつきUCTとDLMCの設計は付録A.2の通りだが，パラメータは思考時間がともに3秒になるよう定められている．PWつきUCTの手番ごとのシミュレーション回数は6000回で，DLMCは200個のサンプリングされた行動に100回ずつシミュレーションを割り当てる．

このとき勝率は表7のようになった．1手番目で攻撃は行われず，2手番目以降は同じ性能のAIが両プレイヤを操作するので，最初の駒の布陣が勝率の偏りに貢献したと考えられる．多様な局面に広く（大雑把な）シミュレーションを割り当てたDLMCが2つの局面で有意に勝率が高い．このように，戦術的TBSでは相手に隙を見せずに序盤の駒配置を行う工夫によって，有意な勝率向上に寄与することがある．

とはいえ，こうした異なる探索が有効に働く局面間の区別をどうつけていくか，また自分の攻撃的手順を詳しく読むべき部分局面と守備的手順をよく読むべき部分局面が組み合わせられたような局面ではどのようにそれらを分割するかも難しい問題として存在している．

8. おわりに

本稿で我々は，内政ルールのないターン制ストラテジーを扱うための対象問題として，戦術的TBSという問題クラスを定め，それが含む課題のうち主要なものを3つ述べた．複数着手性による合法手組合せ数の過剰な増大は，例えば行動の適切な枝刈りまたは状態評価の洗練によってアプローチできる事を示し，初期局面多様性による状態評価の即興性については動的な手法によりアプローチできる可能性を示した．攻撃行動組合せが持つ戦局への影響力の大きさについては，攻撃側と守備側でそれぞれ異なる探索アプローチで高い成果が発揮できる事を示した．これにより戦術的TBSで設計者が出会う課題の特徴をある程度明確化したと考える．

戦術的TBSの分野で人間の上級者より強い性能を発揮できるAIの作成は，我々の知る限りまだ報告されておらず，それらの課題を人間に匹敵する程度には高い精度で解決していく事が現在望まれている．そして戦術的TBSで強いAIを作成するという問題への，考えられる拡張としては，まず占領や生産の要素を加えた条件の追加があると考える．多くのターン制ストラテジーでは（主にゲーム中盤以降で）占領または生産が戦略でかなり重要になってくる初期局面が登場し，そうした場合に高い性能を発揮できるAIの実現には価値がある．ただし，この「占領」および「生産」ルールがゲームに与える影響は単純ではなく，例えば「特定のマスの防衛または放棄の判断」といった全く新しい種類の課題を設計者に課すと予想できる．

また，さらにその先にはルールが複雑な種類のターン制ストラテジー向けの拡張が想定されて，Zocや索敵，地形の多層性などが対象問題に追加すべきルールの候補となる．その中では「索敵」が影響の度合いとして大きそうである．というのも，ゲームに不完全情報性を与え，相手プレイヤの考えを予想することの重要性もゲームに加わるためである．そしてゆくゆくはターン制ストラテジー全般で人間の上級者プレイヤを相手に，初期戦力の不平等などに頼らず，純粋な戦略の工夫で拮抗できるAIプレイヤが実現されればそれは意義のあることだと我々は考える．

付　　録

本稿の実験におけるAIの設計を詳べる．

A.1 実験環境

実験に用いたマシンのスペックを記す．プロセッサはIntel(R) Core(TM) i7-2600 CPU 3.40Ghzで，RAMは8.00GB，OSはWindows 7 Service Pack1の64ビットであり，4コア8スレッドで動作する．本稿で探索は基本的に逐次的に処理されるが，DLMCのみ並列化されている．DLMCにおいて末端のLeaf局面で並列にシミュレーションする．また対戦実験に用いたTUBSTAPのバージョンは1.07である．

A.2 本稿に共通するAIの設定

A.2.1 UCT

本稿の実験で使われるUCT探索の係数Cは1.0であり，1駒の着手を1ノードとした．手番あたりのシミュレーション数（N_{allSim}とおく）は固定値する．与えられた局面にある自軍の行動可能な駒数をn_{myU}とすると，$\frac{N_{allSim}}{n_{myU}}$回のシミュレーションを行って1つの駒を着手する．なお，この$\frac{N_{allSim}}{n_{myU}}$回ずつのシミュレーションでは自軍の駒は様々な順序で動き得る．そのシミュレーションの結果，実際に着手する駒（ルート局面から勝率が最も高いノードに達するために最初に動かす必要のある駒）とその行動が決まる．

このプロセスをn_{myU}回繰り返す事で1手番の行動を終える．シミュレーション中では，攻撃が可能な駒は必ず攻撃行動をとる．また1回のシミュレーションは16手番の着手で打ち切られ，HP差で勝敗の判定に移る．

A.2.2 PW

PWつきUCTもUCT探索部について同様の仕様をベースにするが，訪問対象となるノード数は，n_{sim}回目のシミュレーションで$\lfloor \frac{log\frac{n_{sim}}{40}}{1.4} + 2 \rfloor$個（$n_{sim} \leq 3000$のとき）

または $\lfloor \frac{log \frac{n_{sim}+2000}{45}}{1.2} - 11 \rfloor$ 個 ($n_{sim} > 3000$ の時) となる．本稿でのノードの並べ替え，つまり訪問するノードに加える優先順位については，攻撃行動によるノードを移動のみの行動によるノードより優先する．それ以外に基準はなく，攻撃行動のノード間および移動行動のノード間では，訪問対象に加える順序はそれぞれ完全にランダムに決定される．

A.2.3 AAS

AAS では，攻撃ができない駒，あるいはパラメータ L を超える分の駒は合法手から一様にランダムな行動をとる．局面評価には 4.1 項の状態評価関数を用いる．

A.2.4 DLMC

DLMC の局面評価関数は，I（歩兵）の重みを 0.2，他の駒の重みを 1.0 とする HP の線形和による関数（4.1 項参照）を用いる．またシミュレーション打ち切りの深さは 2 手番分の深さとする．本稿では DLMC のみ並列化されており，ルート局面に対して 1 手番の合法手組み合わせを適用した局面（深さ 1 の局面かつ末端局面）が，並列なランダムシミュレーションで評価される．

参考文献

[1] Campbell, M., Hoane, J.A. and Hsu, F.: Deep blue, Artificial intelligence, Vol.134, No.1, pp.57-83 (2002).

[2] 電王戦：棋戦情報，入手先 ⟨http://www.shogi.or.jp/kisen/denou/⟩（参照 2016-05-29）.

[3] Silver, D., Huang, A., Maddison, J.C., et al.: Mastering the game of Go with deep neural networks and tree search, Nature, Vol.529, No.7587, pp.484-489 (2016).

[4] Mizukami, N. and Tsuruoka, Y.: Building a Computer Mahjong Player Based on Monte Carlo Simulation and Opponent Models, Proc. CIG, pp.275-283 (2015).

[5] SQUARE ENIX : FINAL FANTASY TACTICS 獅子戦争 — SQUARE ENIX，入手先 ⟨http://dlgames.square-enix.com/fft/⟩（参照 2016-02-19）.

[6] システムソフト・アルファー株式会社：製品一覧&購入，入手先 ⟨https://www.ss-alpha.co.jp/products/index.html⟩（参照 2016-02-19）.

[7] Sid Meier's Civilization Beyond Earth，入手先 ⟨https://www.civilization.com/jp/games/civilization-beyond-earth/⟩（参照 2016-02-19）.

[8] Buro, M. and Churchill, D.: Real-time strategy game competitions, AI Magazine, Vol.33, No.3, pp.106-108 (2012).

[9] Ontanon, S., Synnaeve, G., Uriarte, A., et al.: A survey of real-time strategy game AI research and competition in StarCraft, Computational Intelligence and AI in Games, Vol.5, No.4, pp.293-311 (2013).

[10] Turn-based tactics - Wikipedia, the free encyclopedia, 入手先 ⟨https://en.wikipedia.org/wiki/Turn-based_tactics⟩（参照 2016-02-19）.

[11] 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，第 18 回ゲームプログラミングワークショップ，pp.138-145 (2013).

[12] Bergsma, M. and Spronck, P.: Adaptive Spatial Reasoning for Turn-based Strategy Games, Proc. AIIDE, pp.161-166 (2008).

[13] 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，第 19 回ゲームプログラミングワークショップ，pp.32-39 (2014).

[14] 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).

[15] コーエーテクモゲームス：「信長の野望」30 周年記念サイト，入手先 ⟨http://www.gamecity.ne.jp/nobunaga30th/⟩（参照 2016-02-19）.

[16] Wender, S. and Watson, I.: Using reinfocement learning for city site selection in the turn-based strategy game Civilization IV, Proc. CIG, pp.372-377 (2008).

[17] Amato, C. and Shani, G.: High-level Reinforcement Learning in Strategy Games, Proc. AAMAS, pp.75-82 (2010).

[18] Branavan, K.R.S., Silver, D. and Barzilay, R.: Learning to win by reading manuals in a monte-carlo framework, Proc. ACL, pp.268-277 (2011).

[19] Ulam, P., Goel, A., Jones, J., et al.: Using Model-Based Reflection to Guide Reinforcement Learning, Proc. IJCAI, (2005).

[20] Houk, A.P.: A Strategic Game Playing Agent for FreeCiv, Master thesis, Northwestern University (2004).

[21] Hinrichs, R.T. and Forbus, D.K.: Analogical Learning in a Turn-Based Strategy Game, Proc. IJCAI, pp.853-858, (2007).

[22] Perla, P. P.: The art of wargaming: a guide for professionals and hobbyists, Naval Institute Press, Annapolis (1990). 井川宏（訳）：無血戦争，ホビージャパン (1993).

[23] Hoki, K. and Kaneko, T.: Large-Scale Optimization for Evaluation Functions with Minimax Search, Journal of Artificial Intelligence Research, Vol.49, pp.527-568 (2014).

[24] Kocsis, L. and Szepesvari, C.: Bandit based monte-carlo planning, Machine Learning: ECML, pp.282-293 (2006).

[25] Auer, P., Cesa-Bianchi, N. and Fischer, P.: Finite-time analysis of the multiarmed bandit problem, Machine learning, Vol.47, pp.235-256 (2002).

[26] Kloetzer, J.: Monte-Carlo Techniques: Applications to the Game of the Amazons, Ph.D. thesis, Japan Advanced Institute of Science and Technology (2010).

[27] 山本一成，竹内聖悟，金子知適，田中哲朗：コンピュータ将棋における Magic Bitboard の提案と実装，第 15 回ゲームプログラミングワークショップ，pp.42-48 (2010).

[28] Fujiki, T., Ikeda, K. and Viennot, S.: A platform for turn-based strategy games, with a comparison of Monte-Carlo algorithms, Proc. CIG, pp.407-414 (2015).

[29] Nan, H., Fang, B., Wang, G., et al.: Turn-Based War Chess Model and Its Search Algorithm per Turn, International Journal of Computer Games Technology, Vol.2016, Article ID 5216861 (2016).

[30] Knuth, E.D. and Moore, W.R.: An Analysis of Alpha-Beta Pruning, Artificial Intelligence, Vol. 6, No. 4, pp. 293-326 (1975).

佐藤 直之 （学生会員）

2012 年京都大学工学部電気電子工学科卒業．同年より，北陸先端科学技術大学院大学情報科学研究科博士前期課程に在学中．ゲーム AI に関する研究に従事

藤木 翼 （学生会員）

2013 年岐阜工業高等専門学校本科電気情報工学科卒業．同年より，北陸先端科学技術大学院大学博士前期課程に在学中．ゲーム AI に関する研究に従事

池田 心 （正会員）

1975 年生．1999 年東京大学理学部数学科卒業．2000 年東京工業大学総合理工学研究科知能システム専攻修士課程修了．2003 年同博士課程修了，博士（工学）．同年京都大学メディアセンター助手，2010 年北陸先端科学技術大学院大学情報科学研究科准教授．ゲーム，進化計算，パズル等の研究に従事．計測自動制御学会，進化計算学会，情報処理学会等会員．コンピュータ囲碁フォーラム理事．