Vol.2022-GI-48 No.7
2022/7/2

ターン制戦略ゲームにおける攻撃優先プレイアウトの影響

奥田真[1,a)]　池田心[2]　橋本剛[1]

概要：ターン制戦略ゲームの AI は TUBSTAP というプラットフォームで研究が行われているが，AI の強さは初心者プレイヤと互角程度に留まっている．その理由の 1 つとして，TUBSTAP の複数着手性が挙げられる．1 手で複数の駒を動かすルール上，合法手が多すぎて深く読めなくなるのである．このため，TUBSTAP ではモンテカルロ木探索が有望とされ，枝刈りなどノードを減らす工夫を行う研究がされてきた．一方でノードを減らす以外の工夫としてプレイアウトの方策改良があるが，TUBSTAP でこれを検討した研究は少ない．TUBSTAP における従来のプレイアウト方策には，駒の移動行動と攻撃行動を事前に決めた確率で選択するものがあるが，本論文ではこの確率を攻撃行動を優先するように変えたプレイアウト方策を提案する．対戦実験の結果，提案した方策は従来の方策に対しあるマップで勝率 87.7%で大幅に勝ち越し，提案方策は有効であることが分かった．また，複数マップでの対戦実験により提案方策の影響を調査した．

キーワード：TUBSTAP，MCTS，UCT，プレイアウト，ターン制戦略ゲーム

The Impact of Attack Priority Playout in Turn-Based Strategy Games

Okuda Makoto[1,a)]　Ikeda Kokolo[2]　Hashimoto Tsuyoshi[1]

Abstract: AI for turn-based strategy games has been studied on the TUBSTAP platform. However, the strength of the AI has remained at about the same level as that of novice players. One of the reasons for this is the multiple-move nature of TUBSTAP. Due to the rule of moving multiple pieces in one move, there are too many legal moves to read deeply. For this reason, Monte Carlo tree search is considered promising for TUBSTAP, and research has been conducted to reduce the number of nodes by pruning branches and so on. On the other hand, there is another way to reduce the number of nodes, which is to improve the playout strategy, but few studies have examined this in TUBSTAP. The conventional playout strategy in TUBSTAP is to select a piece's move or attack action with a predetermined probability, but in this paper, we propose a new strategy to select a piece's move or attack action with a predetermined probability. In this paper, we propose a playout strategy that changes this probability to give priority to attacking actions. In a competitive experiment, the proposed strategy significantly outperformed the conventional strategy on one map by 87.7 percent, and the proposed strategy was found to be effective. We also investigated the impact of the proposed strategy in a competitive experiment on several maps.

Keywords: TUBSTAP, MCTS, UCT, Playout, Turn Based Strategy Game

1. はじめに

強さを求めるゲーム AI がプロに勝利する瞬間は，コンピュータ技術発展の象徴として世界中の人々から関心が向けられている．1994 年にチェスで Deep Blue が，2016 年には囲碁で AlphaGo が世界チャンピオンに勝利した [1][2]．次の段階として，よりルールが複雑なゲームでも強い AI を作ることに注目がされている．

ターン制戦略ゲームの AI は，TABSTAP（TUrn Based STrategy Academic Package）[3][4] というプラットフォー

[1] 松江工業高等専門学校
National Institute of Technology,Matsue College
[2] 北陸先端科学技術大学院大学
Japan Advanced Institute of Scienceand Technology
[a)] s2106@matsue-ct.ac.jp

ムで研究が行われている．しかし，TUBSTAP の AI の強さは初心者プレイヤと互角程度に留まっており，その理由の 1 つとして TUBSTAP の複数着手性 [5] による合法手の多さが挙げられる．1 手で複数の駒を動かすルール上，合法手が多すぎて強い手を発見しにくくなるのである．このような性質を持つ TUBSTAP において，強い AI を作るために広く用いられている探索手法が MCTS（Monte Carlo Tree Search）である．MCTS は PlayOut（以下，PO）と呼ばれるランダムなシミュレーションによって，ゲームを終局まで進めてその勝率をノード評価に用いる．ノード評価に評価関数を必要としないため，評価関数が作りにくい TUBSTAP に適しているといえる．大会でも MCTS の代表的アルゴリズムである UCT（Upper Confidence bounds applied to Trees）探索をする AI が高い戦績を残しており [6], その有望さを示唆している．

TUBSTAP で UCT 探索を行う AI は，枝狩りなどノード数を減らす方向性で研究が行われている [7]．一方で，ノード数を減らす以外の工夫として PO の改良が挙げられるが，これを TUBTSAP で検討した研究は少ない．囲碁では PO で完全にランダムな手を選択するよりも，一定の着手確率（方策）に従って手を選択する方が有力とされ研究も行われており [8][9][10][11][12]，TUBSTAP でも PO 方策を検討することは重要な問題だといえる．

そこで本論文では，TUBSTAP における UCT 探索の PO 方策を改めて検討し，従来方策より良い方策として攻撃を優先する PO を提案する．また，複数種類のマップでの対戦実験を行うことで，提案方策の影響を調査する．

提案方策の攻撃優先 PO を用いた AI は，2022 年の TUBSTAP 大会で準優勝し [13]，現行 AI の中でもある程度の強さを持っていることを確認している．

まずは前提知識として 2 章で TUBSTAP の特徴とルールを簡単に解説してから，3 章で関連研究を説明する．そして 4 章で PO 方策の検討をして，5 章で提案方策である攻撃優先 PO の影響を調査する．そして 6 章でまとめの考察を行う．

2. TUBSTAP のルールと特徴

TUBSTAP[3] はルールが複雑なターン制戦略ゲームを汎化・簡略化することで学術研究しやすくしたプラットフォームである．本章では TUBSTAP のルールと特徴を簡単に解説する．

2.1 TUBSTAP の簡単なルール

TUBSTAP では，図 1 に示すユニットを将棋のようにターンごとに動かして戦わせる．戦車や歩兵などのユニット種があり，移動可能範囲も異なる．例として，図 1 の歩兵 1 の移動可能範囲はオレンジの点線で囲まれた 9 マスの領域である．将棋と同様のターン制ゲームではあるが，多くの相違点も存在する．把握しておくべき相違点を表 1 に示す．また，本論文に関わる 3 つのルールのみ詳細に紹介する．

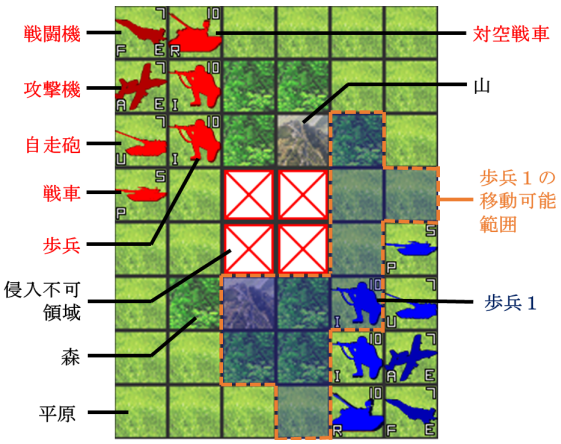

図 1 TUBSTAP のユニットとマップの一例
Fig. 1 An example of TUBSTAP units and maps

表 1 TUBSTAP と将棋の相違点
Table 1 Differences between TUBSTAP and Shogi

ルール	将棋	TUBSTAP
1 手に動かす駒	1 駒	全自軍ユニット
駒の HP	×	○
駒の行動	移動	移動と攻撃
敵駒の取り方	重ねる	攻撃で HP を 0 にする
駒の反撃	×	○
駒の相性	×	○
勝利条件	王をとる	（本文で説明）
初期駒配置と盤	固定	可変
地形効果	×	○

- 行動の種類
  「移動」と「攻撃」の 2 種類の行動があり，移動は移動可能範囲に移動し，攻撃は移動可能範囲に移動後 4 近傍に隣接する敵ユニットを攻撃する．攻撃は移動も含んでいるが，本論文では「移動＋攻撃」ではなく「攻撃」と まとめて表記することに注意されたい．
- 勝利条件
  TUBSTAP の勝利条件は相手を全滅させることのほかに，マップごとに規定されたターン数（以下，ターン制限）に到達した際の判定勝ちがある．本論文ではこれを Turn Over Death（以下，TOD）と表記する．ターン制限に到達しても両軍ともに全滅していない場合，TOD の判定に入る．自軍と敵軍の合計 HP の差を比較し，より多い方のプレイヤが判定勝ちとなる．ただし，合計 HP の差が規定されたしきい値（以下，TOD 判定しきい値）以下だった場合，引き分けとし

て判定する．

- ユニット間の相性
  ユニットを攻撃する際に相性によって与えるダメージが異なる．例えば歩兵ユニットは戦車ユニットに効果的な攻撃はできず，逆に戦車は歩兵の HP を大きく削る攻撃が可能である．

2.2 複数着手性による組み合わせ爆発

TUBSTAP は将棋と異なり 1 駒だけでなく 1 ターンに全自軍ユニットを動かす．これを複数着手性 [5] と呼び，展開されるゲーム木のノードは全自軍ユニット行動の組み合わせとなる．ここで注意したいのがユニットを動かす順序である．例えば味方ユニットが既にいるマスには移動することができないため，順序によりユニットの移動可能範囲が変わってくる場合があるのだ．このように行動組み合わせに至るまでの順序も考慮する必要があり，ノード数は爆発的に増える．これをユニットの組み合わせ爆発 [14] と呼び，膨大な計算量の原因となっている．

複数着手性による組み合わせ爆発の事例として，初ターンの合法手を概算してみる．例えば図 1 で，青軍のユニット 7 体を操作する場合の合法手を考える．ここで 1 ユニットの移動可能範囲を歩兵 1 の 9 マスで近似する．[*1] このとき 7 体のユニットが各々の移動可能範囲 9 マスのうち，被らないようそれぞれ異なるマスに移動した場合，9^7 通りの行動組み合わせが考えられる．1 つの行動組み合わせにつき，ユニット数の階乗通りの順序も考える必要があるため，$9^7 \times 7! \fallingdotseq 2.41 \times 10^{10}$ 通りの合法手が考えられる[*2]．将棋の初手の合法手が 30 通り，囲碁が 362 通り [15] であることから，TUBSTAP は初手以降の合法手も将棋や囲碁より多いことは明らかである．このように，複数着手性による組み合わせ爆発が TUBSTAP というゲームの特徴といえる．

3. 関連研究

3.1 TUBSTAP 以外のゲームにおける PO 方策

TUBSTAP 以外のゲームでは PO 方策の検討が進んでおり，その知見を TUBSTAP にも活かすため，関連研究をいくつか紹介する．紹介した PO 方策を本論文で導入するわけではないが，TUBSTAP とそれ以外のゲームでは PO 方策検討の進み具合に大きな開きがあることを示すため，可能な限り幅広い種類の方策を取り上げる．

3.1.1 囲碁

囲碁で MCTS を実装した，Rémi Coulom による AI「CrazyStone」は手のパターンを評価する Elo レーティングにより合法手の確率分布を作り，PO ではその確率分布に従って手を選択する方策をとっている [8]．また，合法手の確率分布を最適化するシミュレーションバランシングと呼ばれる手法もあり，局面の minimax 値を用いて PO 時の勝率を最大化するような方策をオンライン学習させている [9][10]．これらの方策は強いパターンを棋譜から学習させる必要があり，棋譜の蓄積がない TUBSTAP ではこれらの方策の導入は難しいだろう．

一方で，既に出た PO の結果を似た局面の評価に利用する RAVE（Rapid Action Value Estimation）[11], PO 時に勝った手を記録しておいて再び似た局面が現れたら記録した手を打つ方策 LGRF（Last Good Reply with Forgetting）[12] といった方策は教師データを必要としないため，TUBSTAP への応用が期待できる．

PO をしない AI に Deepmind の「AlphaZero」がある [16]．PO の結果を再現するようなニューラルネットワークを PO の代わりに用いるため，一切のヒューリスティクスを必要としない．このアプローチは RAVE,LGRF 同様に TUBSTAP への応用が期待できるが，PO を行わないため本論文では取り扱わないものとする．

3.1.2 将棋

将棋における PO はランダムな手を打つ方策だと終局しにくい問題点がある．これを解決するために橋本らは PO の着手を 10 手程度に制限し，末端局面で評価関数を使用して MCTS を実現した [17]．ここから佐藤らは CrazyStone 同様 Elo レーティングを用いることで PO の終局確率を向上させている [18]．しかし PO の速度が遅くなる問題が残り，これに対し宇賀神らは遷移確率のみを用いるシンプルな方策により解決を図っている [19]．宇賀神らは，PO の速度は MCTS における重要な要素であると述べ，方策で取り扱う特徴を絞り込み，PO 回数を稼ぐ方向性で提案を行っている．将棋における PO 方策には TUBSTAP に直接応用できそうなものは見当たらなかったが，PO の速度は重要であると示唆された．

3.1.3 ガイスター

不完全情報ゲームのガイスターでも PO の改良が試みられている．従来のガイスターの PO は完全にランダムな手を選ぶため，実際には取らないような手をとってしまう問題があるとされる．これに対し栃川らは遺伝的プログラミングを用いて方策の質を向上させることで解決を図っている [20]．この手法は駒の残数などいくつかのヒューリスティックな条件を用意し，遺伝的操作を適用して生き残った条件で方策を作成する．この方策は用意する条件に性能が依存するため，適当なヒューリスティックな条件を用意することが課題となる．TUBSTAP は定石も存在せず，棋譜の蓄積もないため，条件を用意することは難しく導入ハードルは高いだろう．

[*1] 正確には歩兵の 2,3 倍移動可能なユニットもいるので，小さめの見積もりとなる．
[*2] 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 $n^r \times r!$ 通りになる [14]．これらのほとんどは同じ状態に行き着く．

3.2 TUBSTAP における従来の PO 方策

TUBSTAP における PO 方策を検討するにあたり，従来の PO 方策を解説する．TUBSTAP で採用されている PO 方策は 2022 年 5 月現在，2 種類確認される．

3.2.1 静的な確率で行動選択する方策

藤木らが提案した深さ限定モンテカルロ法 [21] では，移動と攻撃の行動のうち，移動を 20%，攻撃を 80%の確率で選択する方策を採用している．事前に決めた静的な確率で行動選択するため，本論文ではこの方策を静的確率 PO と呼ぶ．藤木らは静的確率 PO について，攻撃時ユニット間の相性を考えていないため，これを解消すれば性能の向上が見込めるとしている．また，TUBSTAP の PO 方策そのものについては，移動と攻撃を完全なランダムで選択するとなかなか終局しなくなるとし，かといって攻撃を優先させると，相手の安易な攻撃を期待して待ちの行動が多くなると述べている．しかし，藤木らの論文はあくまで深さ限定モンテカルロ法という探索手法がメインテーマであるため，これらの論述を裏付ける PO 方策の実験的検討などは行っていない．

静的確率 PO は，武藤らが提案したユニット行動木を UCT 探索する AI「M-UCT」にも引き継がれており [22]，提橋らによって枝狩りやユニット行動木の多段化といった工夫をする研究がされている [7][23]．しかし武藤ら・提橋らの論文でも，静的確率 PO を引き継いだ理由など，PO 方策に関わる記述はない．

3.2.2 動的な確率で行動選択する方策

2019 年の大会で準優勝した坪倉による AI「miTaRashi」では，動的な確率で行動選択する方策を採用している [24]．特に報告がされた文書はないものの，公開されているソースコードより，とれる全行動の評価値を比重としたルーレットを回して行動決定する方策だと考えられる．評価値は（攻撃で敵ユニットに与えるダメージ）-（反撃で敵ユニットに与えられるダメージ）で決定されており，大きいダメージが与えられ，かつ反撃で被ダメージが少ない手が選択されやすい．本論文ではこの方策を動的確率 PO と呼ぶ．動的確率 PO は 2022 年の大会で優勝した，竹内による AI「KUT_GI_UCT」でも採用されている [13]．

4. PO 方策の検討

3.1 章と 3.2 章で述べたように，TUBSTAP 以外のゲームでは PO 方策が研究されているのに対し，TUBSTAP ではまだ PO 方策の検討はあまり進んでいない．そこで本論文では，静的確率 PO の実験的検討から始める．TUBSTAP 以外のゲームからの PO 方策導入や動的確率 PO も検討の対象とすべきだが，まずは一番シンプルな方策である静的確率 PO から始め，その知見を元にそのほかの方策の検討をした方が建設的であると考えた．

4.1 M-UCT

M-UCT は武藤らによって提案された AI で，ユニット行動木と呼ばれるゲーム木で UCT 探索を行うことが最大の特徴である [22]．静的確率 PO を採用した AI の中で新しく，かつ枝刈りなどを行わないシンプルな UCT 探索をするため検討対象として採用した．

4.1.1 M-UCT の PO 方策

M-UCT の PO 方策は静的確率 PO を採用しており，移動と攻撃の行動のうち，移動を 20%，攻撃を 80%の確率で選択する．80%の確率で攻撃行動が選択された場合，全ての攻撃行動の中からランダムに手を選択する．20%の確率で移動行動が選択された場合，ユニットをランダムに選んで，ランダムなマスへ移動する．

4.1.2 ユニット行動木

2.2 章で述べたように，TUBSTAP では合法手が多いため有望な手を見逃す危険性が高い．そこで M-UCT ではゲーム木のノードを全ユニットの行動から，1 ユニットの行動だけにすることで，組み合わせ爆発の影響を抑えている．このときのゲーム木をユニット行動木といい，M-UCT はこの木に対し UCT 探索を行っている．図 2 は ABC という 3 つのユニットがいる局面におけるゲーム木とユニット行動木の例である．簡単化のため各ユニットが 1 種類の行動しかとれないものとすると，ノードが全ユニットの行動の場合は，その組み合わせの 3!=6 個のノードが 1 段に展開する．ユニット行動木の場合は A を展開してから B,C のことを考えるので，1 ターンで複数段の木を展開する．末端ノード数は変わらないものの，1 段目のノード数を見てみるとユニット行動木の方が 3 個少ない．したがって UCT でノードを展開するとき，分岐を少なくすることができる．しかしそれでも合法手の数は膨大なままとされており [22]，2.2 章で述べた特徴は残っている．

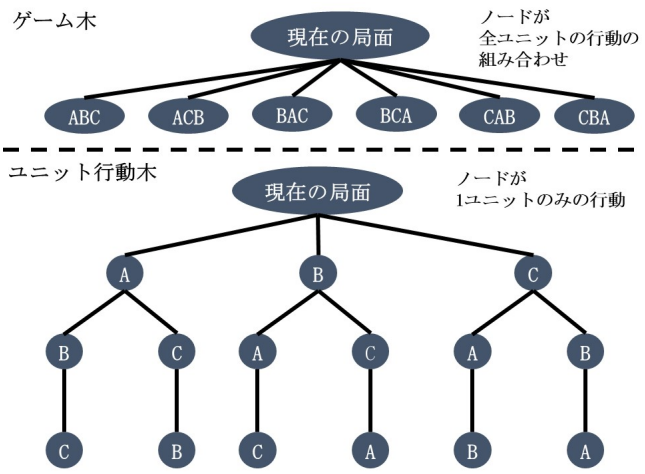

図 2 ユニット行動木の例
Fig. 2 Example of unit action tree

4.2 検討実験　行動選択確率変更の影響

M-UCT の行動選択確率を変更させた場合，勝率がどのように変化するか調査し検討を行う．具体的には表 2 に示す 5 種類の定数確率 PO を用意し，従来手法 M-UCT との対戦実験の勝率を観察する．ここで定数確率 PO を適用した AI の名前は，移動を 20%，攻撃を 80%の行動選択確率なら M20-A80 のように確率値に基づいて表記している．[*3]

表 2　行動選択確率に対応した AI 名と特徴
Table 2　AI names and characteristics corresponding to action selection probabilities

AI 名	行動選択確率% 移動	攻撃	特徴
M0-A100	0	100	攻撃優先
M20-A80	20	80	やや攻撃優先
M50-A50	20	80	半分の確率で攻撃
M80-A20	80	20	やや移動優先
M100-A0	100	0	移動のみ

4.3 検討実験の条件

各種実験条件を述べる．PO 方策以外の差異を設けないようにするため，パラメータは M-UCT に実装してあったものをそのまま用いている．対戦形式は実験時間が長くなりすぎず，かつ勝率も充分に収束するようなバランスのよい回数として，300 回を採用している．また，マップは TUBSTAP にデフォルトで実装されているマップの中で最もシンプルな基本マップ（図 3）とした．

- マシンスペック
  - CPU：AMD Ryzen 5 1600X
  - RAM：16GB
- TUBSTAP のバージョン
  - version 0.108
- M-UCT のパラメータ
  - 1 ターンの制限時間：9700ms
  - UCB 値の定数値：0.15
  - 子ノードの展開閾値：10
  - 1 行動あたりのプレイアウト回数：20000 回
- 対戦形式
  - 対戦回数：300 回
  - 試合の半分（150 回）で先手後手を入れ替える
  - HP と初期位置のランダム変化なし
  - 引き分けは 0.5 勝としてカウント
- 対戦マップ
  - 図 3 に示す基本マップ

[*3] M-UCT の定数確率 PO は移動を 20%，攻撃を 80%の確率で行動選択するので，M20-A80 とは M-UCT を指す．

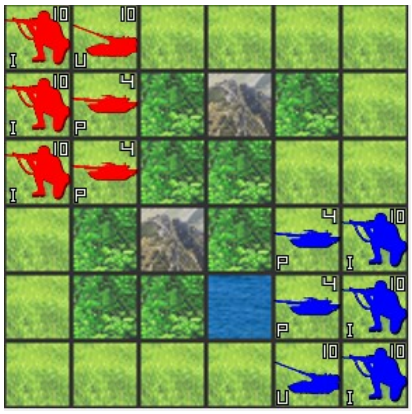

図 3　基本マップ
Fig. 3　Basic Map
ターン制限：29，TOD 判定しきい値：10

4.4 検討結果と提案

自己対戦実験の結果を表 3，図 4 に示す．図 4 の横軸は攻撃確率である．

図 4 より，攻撃確率が上昇するほど M-UCT に対する勝率が向上している．逆に言えば移動確率が上昇するほど勝率が低下しているともいえる．特に M0-A100 は勝率 87.6%で大幅に勝ち越しており，PO 方策の行動選択確率を変えるだけの簡単な工夫で大幅な強化ができたことを意味する．また，このような結果になった理由として PO 回数の増加が考えられる．移動を優先する PO より，攻撃を優先する PO の方が少ないターン数でゲームが終了する．少ないターン数で 1 回の PO が終わるということは，余った時間をさらなる PO に当てることができる．逆に移動を優先する PO ではなかなかゲームが終了しないため，PO 回数を稼ぐことができなくなったと予想される．

以上の検討を経て，本論文ではより良い方策として M0-A100，「攻撃優先 PO」を提案し複数マップにおける影響を調査する．

表 3　行動選択確率変更時の勝率
Table 3　Win rate when action selection probability is changed

AI 名	行動確率% 移動	攻撃	M-UCT に対する勝率
M0-A100	0	100	87.6%
M20-A80	20	80	47.7%
M50-A50	20	80	27 %
M80-A20	80	20	12 %
M100-A0	100	0	7.0 %

5. 実験

5.1 複数マップにおける攻撃優先 PO

攻撃優先 PO の複数マップにおける影響を調査するため，提案方策 M0-A100 と M-UCT との対戦を複数種類のマップで行った．図 5-8 に示すマップをそれぞれ海路・広

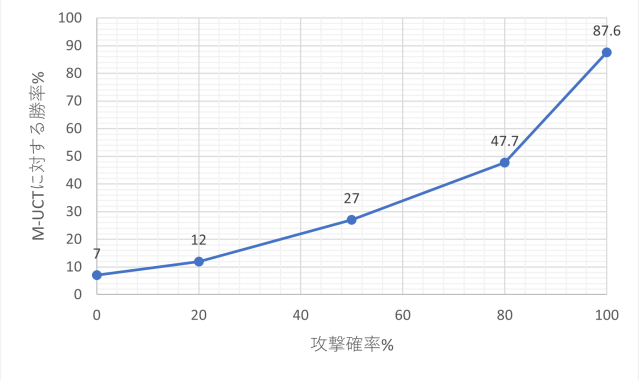

図 4 行動選択確率変更時の勝率
Fig. 4 Win rate when action selection probability is changed

域・隘路・分断マップと呼ぶことにし，この 4 種のマップを対象とした．実験条件は 4.3 章と同じである．表 4 に基本マップも含めた対戦実験の結果を示す．

また，TOD による判定勝ち・判定負けについては勝敗の具体数を省略し，考察に必要な以下の 2 つの値のみ示す．

- TOD 判定数 [回]
  先攻 150 回・後攻 150 回の対戦のうち，TOD 判定が起こった対戦回数
- TOD 率%
  全対戦のうち，TOD 判定数の割合

表 4 複数マップにおける攻撃優先 PO の勝率
Table 4 Win rate of attack priority PO in multiple maps

マップ	着手順序	勝	負	引	勝率	TOD 判定数	TOD 割合
基本	先攻	141	6	3	95.0%	4	2.7%
	後攻	115	24	11	80.3%	12	8.0%
	総合	256	30	14	87.7%	16	5.3%
海路	先攻	103	44	3	69.7%	71	47.3%
	後攻	89	54	7	61.7%	64	42.7%
	総合	192	98	10	65.7%	135	45.0%
広域	先攻	74	67	9	52.3%	25	16.7%
	後攻	128	16	6	87.3%	17	11.3%
	総合	202	83	15	69.8%	42	14.0%
隘路	先攻	76	59	15	55.7%	150	100.0%
	後攻	66	78	6	46.0%	149	99.3%
	総合	142	137	21	50.8%	299	99.7%
分断	先攻	121	13	13	86.7%	50	33.3%
	後攻	69	35	46	61.3%	69	46.0%
	総合	190	48	59	73.9%	119	39.7%

6. 考察

6.1 複数マップにおける攻撃優先 PO

表 4 より，全てのマップで攻撃優先 PO は従来方策に対し総合で勝ち越した．しかし隘路マップの勝率は 50.8%と他マップに比べ低く，隘路マップでは効果が落ちていると思われる．

隘路マップでは攻撃手を多く読んでもあまり意味がなかった可能性がある．隘路マップの TOD 判定数を見ると，300 回の対戦中 299 回で，TOD 割合は隘路マップのみ 99.7%と突出して多い．UCT 探索のアルゴリズムで TOD による決着が多くなったということは，隘路マップではターン制限に到達するような手が強い可能性が高い．ターン制限に到達するような手とはお互いのユニット HP が変動しない移動手であるため，逆に攻撃手の評価値は移動手より低くなったものと推測される．そのため攻撃優先 PO で攻撃手を多く読んでも，その多くが有望ではなく，あまり意味がなかったことが示唆される．

この推論を裏付けるものとして表 4 から，複数マップにおける TOD 率と勝率の相関を図 9 に示す．相関係数は-0.8914 で，おおむね負の相関が認められる．したがって TOD 判定が多いマップが弱点となるといえ，そのようなマップの代表例として隘路マップが苦手であると考えられる．

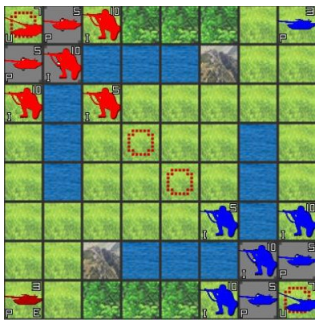

図 5 海路マップ
Fig. 5 Sea Route Map
ターン制限：29
TOD 判定しきい値：0

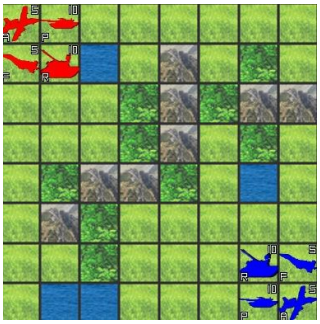

図 6 広域マップ
Fig. 6 Wide-area Map
ターン制限：19
TOD 判定しきい値：3

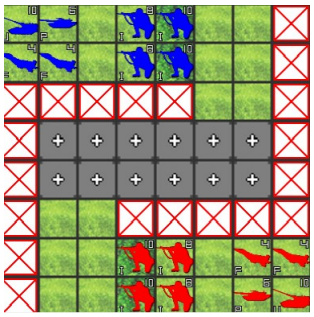

図 7 隘路マップ
Fig. 7 Bottleneck Map
ターン制限：19
TOD 判定しきい値：1

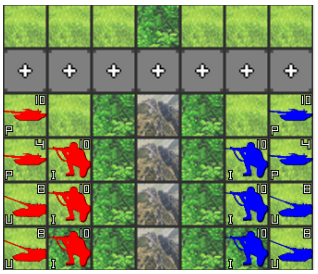

図 8 分断マップ
Fig. 8 Partitioned Map
ターン制限：19
TOD 判定しきい値：10

7. おわりに

本論文では TUBSTAP における UCT 探索の PO 方策に

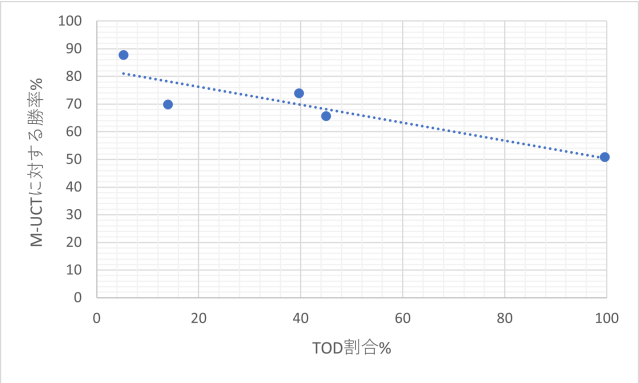

図 9 TOD 率と総合勝率の相関：相関係数-0.8914
Fig. 9 Correlation between TOD rate and win rate: Correlation coefficient -0.8914

ついて検討を行った．検討の結果，従来より良い方策として攻撃優先 PO を提案し，複数種類のマップの対戦実験により提案方策の影響を探った．結果，隘路のあるマップ以外で提案方策の効果が認められ，TOD の判定が起こりやすいマップでは攻撃優先 PO は効果が薄いことが示唆された．課題として，攻撃優先 PO の強さの理由の解明が残った．PO 回数などのパラメータの観察，ベンチマークマップ [25] での対戦を行うことで明らかになるだろう．

参考文献

[1] IBM: IBM100 - Deep Blue (online), available from 〈https://www.ibm.com/ibm/history/ibm100/us/en/icons/deepblue/〉 (accessed 2022-05-30).
[2] DeepMind: AlphaGo (online), available from 〈https://www.deepmind.com/research/highlighted-research/alphago〉 (accessed 2022-05-30).
[3] JAIST 池田研：ターン制戦略ゲーム学術用基盤プロジェクト　TUBSTAP（オンライン），入手先〈http://www.jaist.ac.jp/is/labs/ikeda-lab/tbs/〉(参照 2022-05-30).
[4] 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，pp.146〜153 (2013).
[5] 藤木翼：ターン制ストラテジーのための効率的な探索アルゴリズムの構築，北陸先端科学技術大学院大学　学位論文 (2016).
[6] GAT2021：対戦会 in GAT2021（オンライン），入手先〈http://www.sasebo.ac.jp/~n_sato/competition_2021gat.html〉(参照 2022-05-30).
[7] 提橋凜，西野順二：TUBSTAP におけるユニット別攻撃行動枝刈りの効果，ゲームプログラミングワークショップ 2017 論文集，pp.226〜229(2017).
[8] Coulom, R.: Computing Elo ratings of move patterns inthe game of Go, International Computer Games Association Journal, Vol.30, No.4, pp.198-208 (2007).
[9] Huang S.H., Coulom R. and Lin S.S.: Monte-Carlo Simulation Balancing, ICML (2009).
[10] 渡辺順哉，美添一樹，金子知適：モンテカルロ木探索を統合したプレイアウト方策の最適化，ゲームプログラミングワークショップ 2015 論文集，pp.5〜11 (2015).
[11] Gelly,S., Silver,D.: Combining Online and Offline Knowledge in UCT, Proceedings of the 24th International Conference on Machine Learning, pp.273-280 (2007).
[12] Peter,D.: The Last-Good-Reply Policy for Monte-Carlo Go, ICGA Journal, Vol.32, No4, pp.221-227 (2009).
[13] GAT2022：対戦会 in GAT2022（オンライン），入手先〈http://www.sasebo.ac.jp/~n_sato/competition_2022gat.html〉(参照 2022-05-30).
[14] 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，Vol.57，No.11，pp.2337〜2353，(2016).
[15] 伊藤毅志：コンピュータ囲碁研究の歩み，人工知能学会誌，Vol.27，No.5，pp.497〜500，(2012).
[16] Silver,D., Schrittwieser,J., Simonyan,K., et al.: Mastering Chess and Shogi by Self-Play with a General Reinforcement Learning Algorithm, Nature, Vol.550, pp.354–359 (2017).
[17] 橋本隼一，橋本剛，長嶋淳：コンピュータ将棋におけるモンテカルロ法の可能性，ゲームプログラミングワークショップ 2006 論文集，pp.195〜198 (2006).
[18] 佐藤佳州, 高橋大介：モンテカルロ木探索によるコンピュータ将棋，情報処理学会シンポジウム論文集，Vol.2008，No.11，pp.1〜8 (2008).
[19] 宇賀神拓也，小谷善行：モンテカルロ将棋における遷移確率を用いたプレイアウトの改良，ゲームプログラミングワークショップ 2009 論文集，Vol.2009, No.12, pp.107〜110 (2009).
[20] 栃川純平，竹内聖悟：モンテカルロ木探索ガイスターにおける遺伝的プログラミングを用いたプレイアウト方策の作成，ゲームプログラミングワークショップ 2021 論文集，Vol.2021，pp.124〜129 (2021).
[21] 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，ゲームプログラミングワークショップ 2014 論文集，Vol.2014，pp.32〜39 (2014).
[22] 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，ゲームプログラミングワークショップ 2015 論文集, Vol.2015, pp.54〜60 (2015).
[23] 提橋凜，西野順二：ターン制戦略ゲームにおけるユニット抽象化探索の性能，第 34 回ファジィシステムシンポジウム講演論文集，pp.810〜814 (2018).
[24] GAT2019：対戦会 in GAT2019（オンライン），入手先〈http://www.sasebo.ac.jp/~n_sato/competition_2019gat.html〉(参照 2022-05-30).
[25] 木村富宏，池田心：ターン制戦略ゲームにおけるベンチマークマップの提案，ゲームプログラミングワークショップ 2016 論文集，pp.36〜43 (2016).