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

木村 富宏[1,a)]　池田 心[1,b)]

概要：将棋や囲碁では本来のゲームの部分問題として練習用に詰将棋や詰碁問題群が存在し，初心者の練習用や棋力判定，将棋や囲碁のコンピュータアルゴリズムの性能評価などに用いられてきた．また最適化問題や数理計画法の世界でもベンチマーク問題はしばしば用いられ，アルゴリズムの特定の能力・総合的な能力を誰もが評価できるよう整備されてきた．我々は，ターン制戦略ゲームプロジェクト TUBSTAP においても同様のベンチマーク群が必要であると考える．本稿では，アルゴリズムに必要なさまざまな能力ごとに，難易度の低いものから高いものまでを含むベンチマーク問題群を提案し，既存プログラムでそれらが解けるのかを確認する．

キーワード：ターン制戦略ゲーム，TUBSTAP，ベンチマーク問題，モンテカルロ木探索

Offering New Benchmark Maps for Turn Based Strategy Game

Tomihiro Kimura[1,a)]　Kokolo Ikeda[1,b)]

Abstract: Tsume-shogi and Tsume-go, mating problem of Shogi or Go, are sub-problem of these games. They have been created by many authors, have been played by many players for training, and have been used for evaluating the performance of computer algorithms. Also, benchmark problems are often used in the area of optimization and mathematical programming, for evaluating the specific/total performance of algorithms. We consider such benchmark problems are needed also for TUBSTAP, turn based strategy games. In this paper, we propose many benchmark problems, according to required abilities for these games, and from easy to difficult problems. Finally we show the performance of the existing open-source programs.

1. はじめに

ゲームアルゴリズムの研究の題材として長年の間，将棋や囲碁が取り上げられてきたが，近年では，人間のプロを打ち負かす程のレベルに達している．しかしながら，ターン制戦略ゲームの学術的研究はルールやマップが複雑で，従来の手法をそのまま適用するには探索空間が広大すぎるなどといった事情があるためか，あまり行われていなかった．そのようななか「大戦略」や「ファミコンウォーズ」といった既存のターン制戦略ゲームを分析しつつ，ターン制戦略ゲームに向けた学術研究用基盤プラットフォームとして使用できる「ターン制戦略ゲーム　学術用基盤プロジェクト：TUBSTAP」 [1] が提唱された．

ターン制戦略ゲームでは，囲碁や将棋と異なり，初期局面が固定されておらず，さまざまな初期局面が用いられており，これらはマップと呼ばれている．例えば陸上ユニットしか登場しないマップや航空ユニットしか登場しないマップ，およそ対称なマップから非対称なマップまで多種多様なものがある．

TUBSTAP ではターン制戦略ゲームの基本的なプレイができるようになっているが，本論文投稿時点において用意されている対戦用のマップの数は 10 程度で，十分とはいえない [2]．対戦用マップをアルゴリズムの性能評価に使用する場合，マップごとに駒の価値や取るべき戦略も異なってくることに注意する必要がある．それゆえ，各アルゴリズムにはマップごとに得手不得手があることが頻繁に生じ，性能評価を公平に行うためにはできるだけ多様かつ

[1] 北陸先端科学技術大学院大学 情報研究科

[a)] kt499887@gmail.com
[b)] kokolo@jaist.ac.jp

多数のマップが準備されていることが望ましいと言える．

加えて，いままで使用されてきたマップは概して全探索が困難なほどに大規模であり，アルゴリズムに必要とされる能力が複合的であった．これらのマップは人間が遊んだり，プレイヤの総合的な能力を測るのには問題ないが，「どんな能力が原因で勝率差が出ているのか」「最善手は打てているのか」「最善結果は出せているのか」などを分析することは困難であった．

これらの理由から，本稿では小規模で解析が容易で，必要とされる能力が限定的なベンチマークマップを，多様にかつ多数用意することを目指す．

2. ベンチマークマップの製作ポリシー

TUBSTAP は，大戦略シリーズやファミコンウォーズシリーズを参考に，それらに共通する基本ルールだけを抜き出したゲームおよび開発プラットフォームである [2]．占領や生産といったしばしば使われるルールすら排除されているものの，それでも将棋やチェスに比べれば主に以下の点で複雑かつプログラミングの困難なゲームである．

- 一つの手番で，全ての駒を任意の順序で動かすことができる．また，1 つの駒が移動できる範囲が広い．これらにより可能行動数が指数的に大きくなる．
- 駒の間には相性があり，各駒は特定の駒にしか攻撃できない場合がある．
- 開始局面が固定されておらず，未知の配置が与えられることを想定しなければいけない．駒の種類ごとの数もさまざまであり，そのためマップごとの駒の価値もさまざまである．そのため駒の価値の事前の機械学習は困難である．

これらの結果として，コンピュータアルゴリズムには，「広い探索空間を適切に削減する」「相手の壁役を排除しながら正しい順番で攻撃する」「重要ユニットを同定し，それを壁役で守りながら進軍する」「相手の逃げ道を塞ぐように回り込む，逆に回り込まれないように逃げる」など様々な能力が要求される．これらの総合能力を測るだけであれば，人間プレイヤがよく遊ぶような大規模で対等に近いマップを複数用意すればよいかもしれない．しかし，アルゴリズムの開発あるいは評価のためには，各要素それぞれを評価できるようなマップが整備されていることが望ましい．このようなベンチマーク問題は巡回セールスマン問題 [3] から囲碁 [4]，将棋 [5] などさまざまな分野で用いられている．

本研究では，ベンチマーク問題が備えていると望ましい特徴として以下の点に留意しながら作成を行った．

- その問題で評価したい能力がはっきりとしていること
- 必要のない部分をできるだけ省き，簡素であること
- 人間プレイヤにとっては簡単で，既存プログラムには困難であること
- 同じテーマの問題の中に，難易度の低いものから高いものまであること
- 双方が最善を尽くした結果が解析的に求まること
- 評価を行いたい側ではない側（敵側）の行動を容易に記述できること．そうでないと，自動評価の際に「相手が間違ったので勝てた」ということが生じる．

本論文で紹介するのはこれらのマップの端緒となるものであり，将来的には質・量ともに拡充整理したうえで web ページにて公開する予定である．

3. 既存アルゴリズムの試行

本論文では，ただマップを製作してその意図等を紹介するだけでなく，既存の比較的強いコンピュータプレイヤに実際に問題を解かせることで，現時点でのアルゴリズムの課題を見つけることを目指す．

各マップは，全て評価したい側を先番（赤軍）とし，主に 8 × 8 のサイズを用いた．各マップには“上限ターン”と“HP 差閾値”が個別に設けられており，勝敗は以下のようにして定まる．

- 青軍が全滅すれば，赤軍の勝ち．
- 上限ターンに達し，HP 差（赤軍の HP 合計-青軍の HP 合計）が HP 差閾値より大きければ，赤軍の勝ち．
- 上限ターンに達し，HP 差の絶対値が閾値以下であれば，引分け．
- 上限ターンに達し，HP 差が -閾値未満であれば，青軍の勝ち．
- 赤軍が全滅すれば，青軍の勝ち．

問題によっては，引分けが最善の結果である場合もあることに注意されたい．各アルゴリズム 10 試行を行い，最善の結果を導いた回数をカウントした．

敵側（青軍側）をどのように動かすかは，非常に難しい問題である．なぜなら，詰め将棋や詰碁でもそうであるが，「最善の抵抗」というのは数学的に定義することが困難だからである．詰め将棋であれば詰め手数が最大になる逃げ方が最善の抵抗と定義できるかもしれないが，より面白い手筋・難しい手筋が必要になるのは別の逃げ方の場合であることもある．同様に，TUBSTAP の場合も，手数を最大になるように青軍を操作するのと，赤軍アルゴリズムが間違いやすいように青軍を操作するのでは結果が異なるだろう．これは未解決の課題であるが，本稿では人間が考える最も難しい抵抗を採用した．

用いたアルゴリズムは，2016 年 3 月の UEC-GAT 大会 [6] で上位 3 位までに入賞した Military, M-UCT, M3Lee である．これらのプログラムはソースコードも含め現在 TUBSTAP のパッケージに標準搭載されており，誰でも参考にすることができる．

ただし，これらのプログラムはもともといわゆる対戦用マップ向けに開発されたものであることは明確にしておか

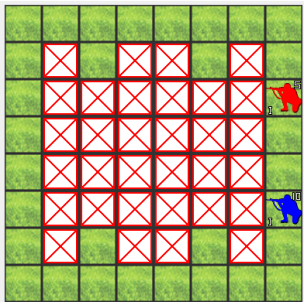

図 1　逃走マップ run_A1

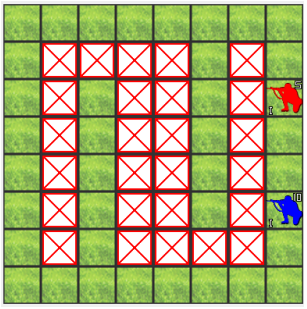

図 2　逃走マップ run_A2

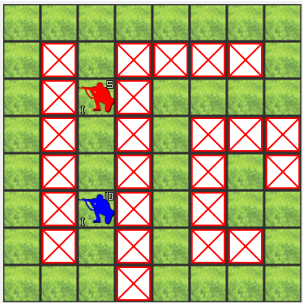

図 3　逃走マップ run_A3

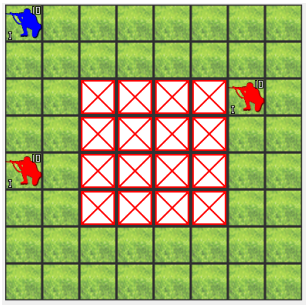

図 4　追跡マップ chase_A1

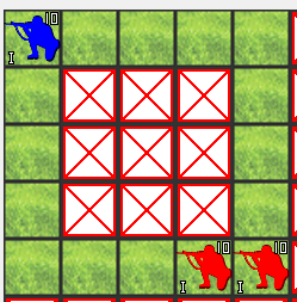

図 5　追跡マップ chase_A2

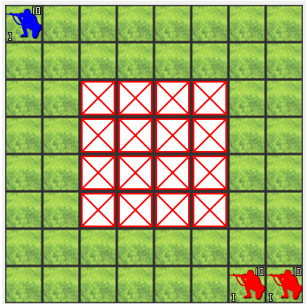

図 6　追跡マップ chase_A3

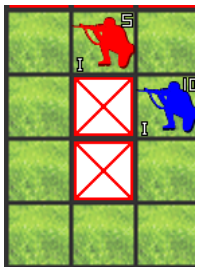

図 7　逃走マップ run_B1

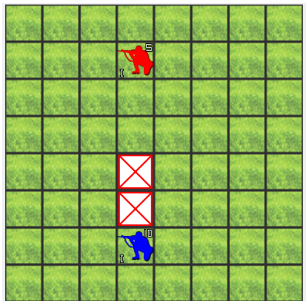

図 8　逃走マップ run_B3

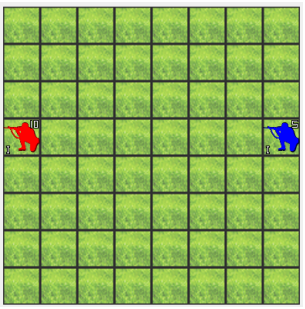

図 9　追跡マップ chase_B1

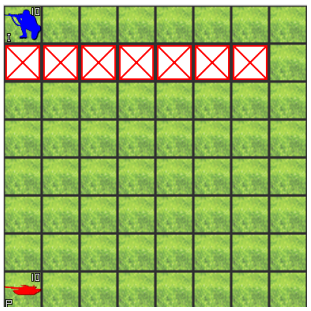

図 10　経路探索マップ path_A1

なければならない．今回紹介するようなマップはある意味で特殊な能力を明示的に要求するため，人間には簡単に解けるようなものであっても，これらのプログラムに解けなかったとしても不思議はない．

さらに，いくつかの問題は「ターン上限まで逃げ切れれば正解」「ターン上限までに倒しきらないと不正解」といったやや通常とは趣の異なる条件を持つため，これに対応していないプログラムも存在する．具体的には，Military はモンテカルロシミュレーションの結果評価として“合計 HP が相手より 1 でも大きければ勝ち”と判定しており，これは「ターン上限までに倒しきらないと不正解」の問題には不適切な対応である．従って，Military については，モンテカルロシミュレーションの結果評価を“合計 HP が相手よりマップで指定された閾値以上大きければ勝ち”に変更して用いた．

4.　追跡・逃走能力検証用マップ

4.1　概要

追跡アルゴリズムはゲームアルゴリズムのなかでも重要かつ基本的なものであって教科書 [7], [8] などでも紹介されることが多い．戦略ゲームの中でも，相手を逃がさないように追跡することが必要な局面は頻繁に出現する．

また，相手の抵抗を考えながら追跡するアルゴリズムを正しく作るには，逃走を正しく行うことも必要である．本稿では，２つの異なるテーマを持つ追跡 (chase) および逃走 (run) のマップセットを紹介する．

chase シリーズでは，赤軍が戦力的に優位な状況にあり，正しく追いつめればターン上限内で青軍を全滅させることができる．ターン上限を超えれば，引分け，すなわち失敗である．

run シリーズでは，赤軍は戦力的に不利な状況にあるが，障害物を利用して正しく逃走すれば，いくらでも逃げ延びることができる．ターン上限に達すれば，引分け，すなわち成功となる．

本章以降，マップの命名規則としては，chase_A1 のように，目的＋_＋テーマ記号＋レベルを用いることにする．

4.2　製作したマップ

表 1 に制作したマップと搭載 AI の成績を示す．どのマップでも HP 差閾値は十分高く設定されており，上限ターン到達時に裁定により勝敗が決まることはない．すなわち，run シリーズならば上限ターンまで逃げ切ればよく，chase シリーズならば上限ターンまでに全滅させる必要がある．

図 1 および 図 2 は袋小路に入らぬように逃げ回れば良いマップで（HP が見にくいことをお詫びする．赤の HP

表 1 追跡・逃走能力検証用マップにおける搭載 AI の成績

マップ名	目的	上限ターン	Military	M-UCT	M3Lee
run_A1	引分け	39	0	0	6
run_A2	引分け	39	0	0	2
run_A3	引分け	19	0	2	0
chase_A1	全滅	19	0	10	9
chase_A2	全滅	19	0	7	10
chase_A3	全滅	19	0	2	0
run_B1	引分け	10	10	1	0
run_B3	引分け	30	0	0	0
chase_B1	全滅	19	4	8	7

表 2 chase_A3 の解析結果

ターン数	9	10	11	12	13	14
ゲーム途中のノード数	127	52	437	15	92	0
同一局面の数	33	11	113	5	29	0
ゲームが終了したノード数	0	289	0	1236	0	290

表 3 経路探索能力検証用マップにおける搭載 AI の成績

Map 名	ターン制限	Military	M-UCT	M3Lee
path_A1	29	0	10	10
path_A2	11	0	9	3
path_A3	39	0	1	2

は 5，青は 10 である），人間ならば容易に引分けに持ち込める．しかし M3Lee が時に正解する以外は逃走に失敗している．途中までうまく逃げても，39 ターンという比較的長期間の間にアルゴリズムの乱数性により失敗するようである．基本的に逃走マップでは相手の行動番をしっかり読む必要があり，それは自分手番だけでも膨大な行動数になる戦略ゲームでは既存アルゴリズムが苦手とするようである．

図 3 はターン数が短く設定されている代わりにマップ右半分方面への遠い逃げ道が袋小路となっており，アルゴリズムが騙されやすい．

図 6 は赤の 2 つの歩兵が二手に分かれて青歩兵を挟み打ちにしなければ勝てないマップである．これも人間であれば挟み打ちの必要性が分かるのだが，コンピュータには難しいマップで殆ど正解されていない．そもそも，run_A1 などの結果から，「片方面だけから攻めた場合，青歩兵が正しく抵抗すれば逃げ切られてしまう」ということが読めていないのが一つの原因であろう．

図 5 および **図 4** は 図 6 を「二手に分かれる」「正しく仕留める」という二つの部分問題に分けたものである．探索空間が小さくなることもあるせいか，これならば完全とは言えないまでも解けるようである．

図 7 はある意味 run_A1 などよりも単純な問題で，相手の裏側に位置して逃げ延びることを目指す．この問題は今まで成績の良くなかった Military が正解しており，アルゴリズムごとに得手不得手があることが分かる．

図 8 は run_B1 の応用問題であり，2 歩下に下がれば同様の方法で逃げ続けることができる．しかしコンピュータは例えば右上方面など，一見安全度が高いと思われる方向に逃げてしまい，最後は追いつめられてしまう．シミュレーションの中ではなかなか正しく青側の追いつめが成功しないことも，右上方面を安全と思ってしまう原因であろう．

図 9 は非常に単純な追いつめ問題であり，run_B3 の部分問題と言える．アルゴリズムは 19 ターンという十分な時間を与えられても必ずしも正解しない．よくある失敗は，敵歩兵が自分から見て（3 歩右，3 歩下）にあるような状況で，下に 3 歩進んでしまうことである．続いて敵歩兵が上に 3 歩進んだときにも上に 3 歩進んでしまえば，これの繰り返しで追いつめられないことがある．

4.3 マップ chase_A3 の全滅ターン数解析

図 6 は赤手番で開始すれば 13 ターンで青を全滅できると，手作業で解析していた．本稿では，青軍の全ての抵抗手段に対して必ず 13 ターンで全滅に至ることを確認するための木探索を行った．

問題設定は少し変更し，青手番で開始することにし，14 ターン内に赤軍が青軍を全滅できることを確認した．解析の手順と条件は，

- 青軍側はすべての逃走経路について展開
- 赤軍側はハサミ打ち戦略で青軍側の逃走経路をせばめていく手順のみを読む．2 つのユニット同時に青軍と戦闘に入り，逃がさないようにする
- 幅優先の MINMAX 木探索を行う．
- 同一ターンで同じ局面ノードが出現したらそのノードは同じノードへと集約する

のように設定した．解析した結果は 表 2 のようになった．ここでは、9 ターン以降のデータを示している．データでは、10 ターン目または 12 ターン目でゲームが終了する場合も多いが、最長でも 14 ターンでゲームはすべて終了した．赤軍側はヒューリスティックを用いて全探索にはなっていないので，今回の結果は「14 ターン以内に全滅はできる」ことを示したものであり，「12 ターンでは全滅できない」ことは示せていない．いずれ，他のマップにも適用できるよう拡張する予定である．

4.4 経路探索能力検証用マップ

追跡能力の発展形として，経路探索能力が必要になる場合もある．現在 TUBSTAP で用いられているような比較的小規模のマップではあまり問題にならないが，生産や占領があるゲームの大規模なマップ，要塞攻略のような趣のあるゲームのマップでは，入り組んだ道筋を辿って目的地に到達する能力が必要である．本節ではこれを経路探索能力検証用マップとして，path シリーズを準備した．

表 3 に製作したマップと搭載 AI の成績を示す．

図 10 は敵歩兵の前に壁があり，回り込んで倒す必要が

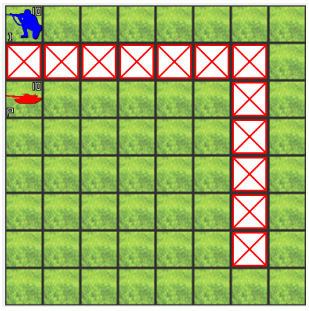

図 11　経路探索マップ path_A2

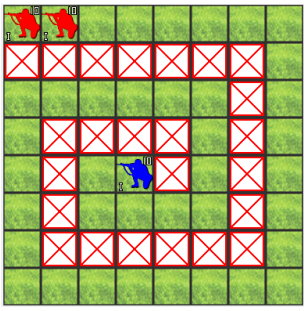

図 12　経路探索マップ path_A3

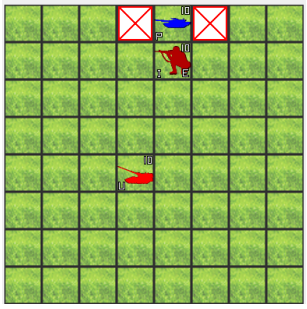

図 13　封鎖マップ block_A1. 歩兵は移動済み

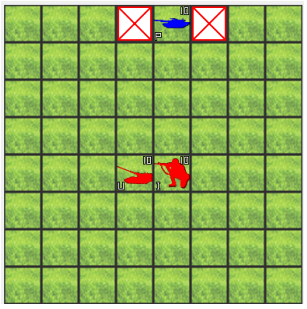

図 14　封鎖マップ block_A2

表 4　封鎖能力検証用マップにおける搭載 AI の成績

Map 名	ターン制限	Military	M-UCT	M3Lee
block_A1	9	0	6	0
block_A2	9	0	3	0
block_A3	9	0	0	0
block_B1	9	10	4	0
block_B2	9	0	0	0

あるマップである．非常に単純であるが，敵との距離をマンハッタン距離で詰めるような移動アルゴリズムの場合，失敗してしまう．

図 11 は遠回りをしなければいけない上に，ターン数上限もぎりぎりに設定されており，比較的難易度が上がっており，成績も若干低下している．

図 12 は歩兵が遠路経由して目的の敵に到達する必要があり，ターン数上限もきつく，既存プログラムではなかなか正解できない．

これらの問題は人間であれば，正解手数を正しく読めるかはともかくとして，正解手順を見つけることは難しくない．特にモンテカルロシミュレーションを使うような手法ではこういった問題は課題であると言える．

5. 封鎖能力検証用マップ

将棋や囲碁でもうまく相手の活動を邪魔するような行動は必要になるが，ターン制戦略ゲームでは駒の移動範囲が広いだけに，周囲を取り囲んで封鎖する，重要地点を押さえて侵攻を免れることは必須の能力になる．

表 4 に製作したマップと搭載 AI の成績を示す．

図 14 は，青戦車の出口を歩兵が犠牲になって足止めし，

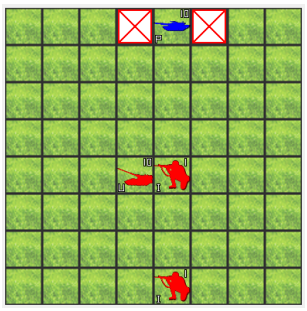

図 15　封鎖マップ block_A3. 歩兵 HP は 1

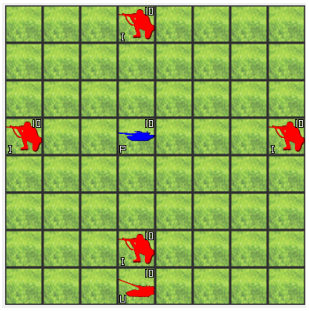

図 16　封鎖マップ block_B1. 歩兵 HP は 10

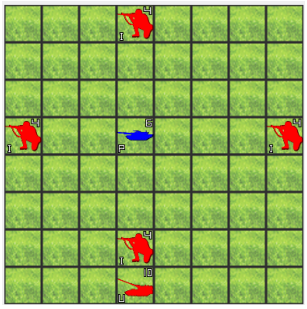

図 17　封鎖マップ block_B2. 歩兵 HP は 4，戦車 6

その間に自走砲が 2 回遠距離攻撃して倒すというテーマのマップである．**図 13** はさらに初手の歩兵の移動を完了させてある簡易バージョンであるが，これでもこの問題を正解するアルゴリズムは少なかった．逃走マップでも述べたように，相手の移動や攻撃を読み切れておらず，マップ下側に逃走して引分けを試みるような挙動が多く見られた．

図 15 は足止めを 2 回にわたり行うテーマのマップであり，この程度であれば人間は簡単に正解を導くが，搭載 AI には解けなかった．

図 16 は，障害物のないマップで歩兵 4 体が戦車の上下左右を封鎖する必要があるマップである．これには Military が全正解したが，その理由は HP10 の歩兵が戦車を攻撃すると HP を 1 減らせるため，「封鎖」ではなく「有効な攻撃」としてこの行動が選択されるためであるようだ．一方 HP 値の異なる**図 17** では，封鎖して 1 回攻撃するだけで（深さ 3 まで読むだけで）正解するにも関わらず，搭載 AI では正解できなかった．

block シリーズに限らず，自走砲の扱いは苦手とする AI が多い印象を受けた．

6. 選択能力検証用マップ

攻撃できる範囲や相手，さらには“行動順”が多いターン制戦略ゲームでは，どの順序でどの相手を攻撃するのかを適切に読む必要がある．select_A,B シリーズでは，正しく行動を選択できれば開始ターンで直ちに敵を全滅できるように設計してある．相手番を読む必要がない代わりに，

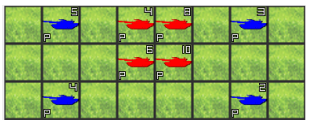

図 18 選択マップ select_A1. 赤軍 HP は 4,6,8,10, 青軍 2,3,4,5

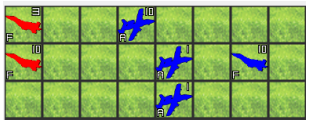

図 19 選択マップ select_C1. 赤戦闘機 HP は 3,10 青攻撃機 HP は 1,1,10

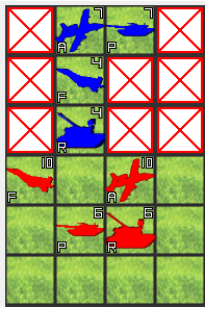

図 20 選択マップ select_B1

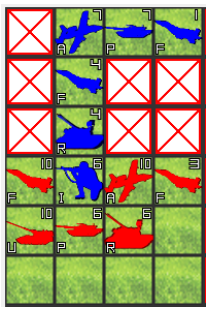

図 21 選択マップ select_B2

表 5 選択能力検証用マップにおける搭載 AI の成績

Map 名	ターン制限	Military	M-UCT	M3Lee
select_A1	1	0	10	2
select_B1	1	0	10	2
select_B2	1	0	2	0
select_C1	9	9	7	10

自分の手番は正確に読む必要があるということである．

表 5 に製作したマップと搭載 AI の成績を示す．

図 18 は，双方 HP の異なる 4 体ずつの戦車を持ち，全ての赤戦車が全ての青戦車に攻撃できる状況である．組み合わせとしては 24 通りの攻撃パターンがあり得るが，正解は 1 通りである．これに M-UCT は毎回正解したが，他のアルゴリズムはほぼ失敗している．

図 20 は，多様なユニットが用いられ，敵を 1 体ずつしか攻撃できないマップとなっている．このような「敵陣を掘っていく」ような読みは実戦では非常に頻繁に生じる．これも M-UCT 以外はほぼ失敗している．**図 21** は同じ趣旨でユニット数が 6 に増えたものであり，人間でも多少迷うかもしれない．M-UCT の正解率も低下している．

図 19 は趣が異なり，「正しく価値の高い相手を攻撃できるか」を調べるものとなっている．ターン制戦略ゲームではマップごとに駒価値が異なるが，本マップでは敵攻撃機は放っておいてもよいために価値が低く，敵戦闘機を攻撃しなければならない．搭載 AI はほぼ正解（引分け）を出したが，場合によっては HP10 の攻撃機のほうを攻撃してしまうようである（負け）．

7. 護衛・タッグ能力検証用マップ

ターン制戦略ゲームでは，特定のユニットを防護しながら進軍することが求められることが多い．例えば占領のために歩兵を戦車で守るであるとか，逆に虎の子の戦車を守るために歩兵が壁になるとか，与えられる役割や必要な戦略は場合によって異なる点が難しくまた面白い．

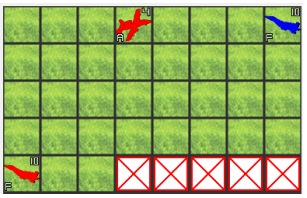

図 22 護衛マップ guard_A1

図 23 護衛マップ guard_B1. HP 左から 6,4,4

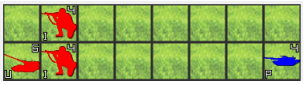

図 24 護衛マップ guard_B2

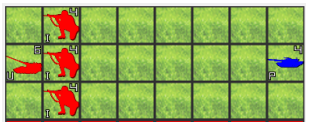

図 25 護衛マップ guard_B3

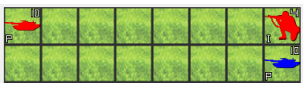

図 26 護衛マップ guard_C1

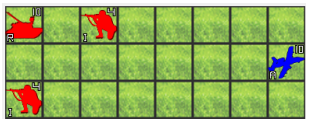

図 27 護衛マップ guard_D1

図 28 タッグマップ tag_A1

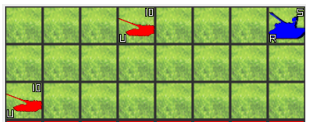

図 29 タッグマップ tag_B1

表 6 護衛･タッグ能力検証用マップにおける搭載 AI の成績．tag_A1 については，ターン上限を 3 から 9 まで変更してそれぞれ評価した．

Map 名	ターン制限	Military	M-UCT	M3Lee
guard_A1	9	9	0	4
guard_B1	19	1	5	1
guard_B2	19	0	0	2
guard_B3	19	0	0	0
guard_C1	19	10	1	10
guard_D1	19	0	0	3
tag_A1	**9**	10	10	10
-	**7**	7	5	6
-	**5**	3	6	6
-	**3**	0	0	0
tag_B1	9	0	0	0

guard シリーズでは，あるユニットを護衛したうえでそのユニットが敵に先制することで勝利に導くようなマップを作成した．

また，単独では敵に先制されたり射程範囲を外されたりしやすい自走砲を，壁役で守るのではなく，複数協調して運用することが必要になるケースもある．これにより，反撃の可能性を見せることで敵の先制を防いだり，攻撃範囲の網を広くかぶせることが可能になる．このような能力を検証するためのマップとして tag シリーズを作成した．

表 6 に製作したマップと搭載 AI の成績を示す．

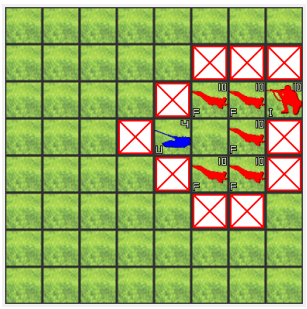

図 30　パズルマップ puzzle_A1

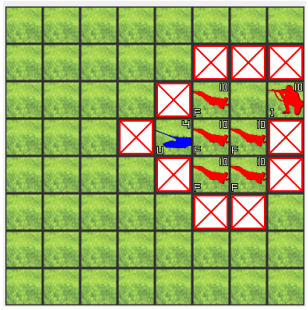

図 31　パズルマップ puzzle_A2

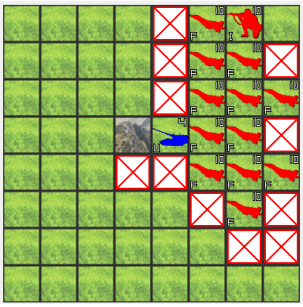

図 32　パズルマップ puzzle_A3

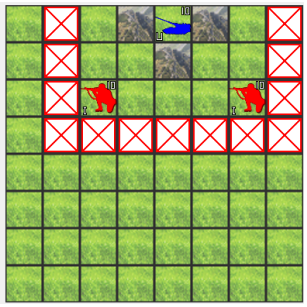

図 33　パズルマップ puzzle_B1

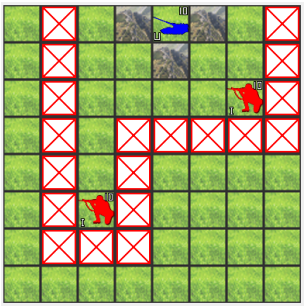

図 34　パズルマップ puzzle_B2

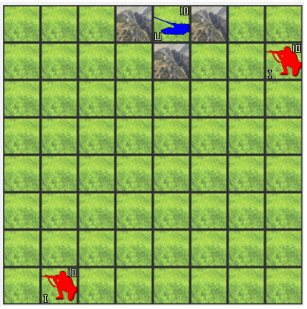

図 35　パズルマップ puzzle_B3

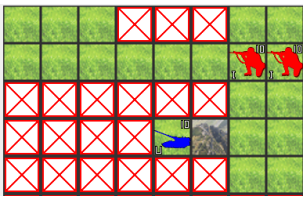

図 36　パズルマップ puzzle_C1

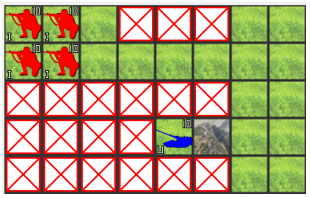

図 37　パズルマップ puzzle_C2

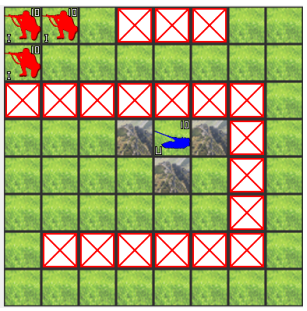

図 38　パズルマップ puzzle_C3

図 **22** は，戦闘機同士のにらみ合いであり，先制した側が勝利する．移動力が同じなので，単独では先制できず手詰まりとなるが，壁役の攻撃機ユニットがあるため，戦闘機が 2 歩右に進み，攻撃機がその直上を護衛すれば勝利することができる．

図 **23** から図 **25** は，歩兵を壁にして自走砲が戦車を追い詰めることをテーマとする．テーマとしては同じであるにも関わらず，マップが広くなるにつれ，探索空間も広くなるためか，搭載 AI の正解率は低下している．

図 **26** は，意味合いとしては guard_A1 と似ており，同種ユニット同士のにらみ合いで相手の進路を妨害することで先制するマップである．搭載 AI の結果も guard_A1 と似た傾向となった．

図 **27** は，guard_A1, guard_C1 をやや発展させたものと言える．対空戦車と攻撃機は移動力が 2 異なるので，1 人の歩兵で対空戦車を護衛することはできない．2 人の歩兵をうまく配置することで，先制することができる．これも搭載 AI には難しかったようである．

図 **28** は，自走砲がマンハッタン距離 2（1 マスナナメや，上下左右 2 マス先）にならないように進行することで，歩兵の逃げ先である死角を作らず全滅できるマップである．正しく操作すれば 3 ターンで青歩兵を全滅させられる．これを搭載 AI に解かせた場合，上限ターン数を 9 とした場合は全 AI が毎回青を全滅させたが，上限ターン数を 7,5,3 と厳しくしていくうちに正解率は下がった．何度か述べている通り，自走砲の扱いは必ずしも上手ではない．

図 **29** は，相手が移動力 5 の対空戦車に変わったものである．これは追いつめ方を間違えると包囲をすり抜けられ，上限ターン 9 では倒せなくなる（最短は 5 ターン）．それなりに正しい先読みが必要で，搭載 AI には解けなかった．

8. パズル型マップ

これまでの問題も，ある意味でパズル性の強いマップであった．本章では，やや必要能力の分類が困難なマップについて紹介する．いずれも，これまで同様，人間には簡単に解けるが，既存のコンピュータには難しいものが多い．

表 **7** に製作したマップと搭載 AI の成績を示す．

図 **31** は，歩兵が敵の自走砲を倒すために，邪魔な戦闘機がスペースを空ける必要がある問題である．select_B シリーズでも攻撃の順番が重要になったが，このマップでは

表 **7**　パズル型マップにおける搭載 AI の成績

Map 名	ターン制限	Military	M-UCT	M3Lee
puzzle_A1	19	10	10	10
puzzle_A2	19	1	10	10
puzzle_A3	9	0	0	0
puzzle_B1	29	0	8	10
puzzle_B2	29	0	2	10
puzzle_B3	29	0	0	9
puzzle_C1	50	0	3	7
puzzle_C2	50	0	0	1
puzzle_C3	50	0	0	0

攻撃だけでなく移動も必要である点が要点である．「攻撃できるユニットから行動する」といったアルゴリズムはしばしば用いられるが，それでは正解が見つからないということになる．**図 30** はこれをより簡単にしたものでこれならば全 AI が正解できるが，puzzle_A2 では正解できなくなる場合がある点に注意したい．

図 32 は，これをさらに難しくした問題であり，人間ならば簡単に解けるが，搭載 AI には解けなかった．

puzzle_B シリーズは，歩兵 2 体で山に囲まれた自走砲に攻撃するというテーマの問題である．片方が犠牲になるという意味では tag シリーズにも関連する．**図 33** は最も簡単なケースで，以降**図 34**, **図 35** と少しずつ自由度が上がったり足並みを揃える行動が必要になったりでやや搭載 AI にとっても難しくなった．

puzzle_C シリーズは，歩兵が自走砲の射程範囲に一斉に突入し，一部のみ攻撃可能位置に辿りつけ勝利できるというテーマの問題である．**図 38** が最も難しく，2 回の同時突入が必要なうえ，目的地までも遠い．これは人間ならば容易に正解できるが，AI には特殊なルーチンを載せなければ無理かもしれない．

図 37 は歩兵数を増やし，目的地までの道のりを短くした簡易バージョンであるが，それでも殆どの場合不正解となってしまった．**図 36** は一回目の一斉突入を終えた状態であり，puzzle_B シリーズに近い状態であるが，それでも正解は稀にしか得られなかった．

8.1 その他のマップ

これまで述べてきたマップのほとんどは，搭載 AI にとっての難しさとはうらはらに，人間にとっては「要点を見抜くこと」「正解を求めること」が容易なものである．一方で，プレイして面白いマップ，人間にとっても難しいマップもまた，拡充していく価値がある．正解の解析が難しいとか，複数の能力が必要になるという意味ではベンチマーク問題とは呼びにくいかもしれないが，そのようなマップが多数整備されていることで，開発者だけでなくゲームそのものの普及にも役立つと考える．

例えば，**図 39** や **図 40** は難しいマップの例である．図 39 は赤軍が戦力的に優れるが，青軍を全滅させようとすれば何度か自走砲に攻撃を受けることになる．青軍は青軍でどこで待ち伏せしてどのように守るのが最善かいくつかの案があり得る．図 40 も赤軍が戦力的に優れるが，直ちに下方に侵攻するのは (7,5) の地形的に不利な場所で青軍に集中砲火を浴びることになるため，左側から回り込まなければならない．何駒が回り込めば引分けを防げるのか，難しい．これらも多数準備する予定である．

9. まとめ

本稿では，ターン制戦略ゲームのためのベンチマーク問題の提案と，既存 AI の評価を行った．人間ならば簡単に解けるような問題でも，既存 AI がうまく解けない場合が多くあることが分かった．

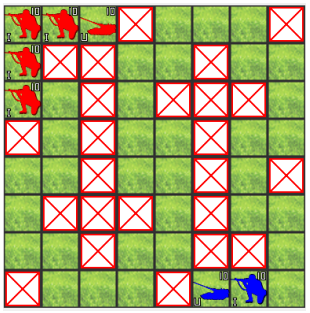

図 39 難しいマップの例 1

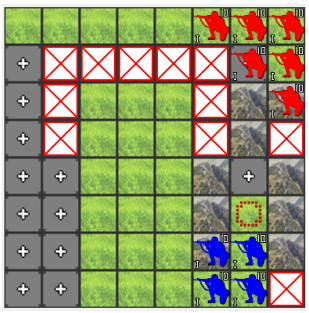

図 40 難しいマップの例 2

今回紹介したマップは，紙面の都合上テーマや結果が面白いもののみである．実際にはより簡単な問題，より難しい問題，ユニットや地形が異なる問題など，さまざまなものを準備していかなければいけない．

また，この手のゲームでは「双方が先攻不利と考え陣を構えたまま膠着する」ことがしばしばあり，AI 開発者およびゲームデザイン・マップデザインの課題となっている．その意味で，「引分けを狙うか，勝ちを目指すか」を迷わせるようなベンチマーク問題も作成していきたい．

今回は優勢勝ち（ターン上限に達したが，HP 合計に閾値以上の差を付けた）を狙うようなマップも作成していないので，これも課題である．合わせて，囲碁などで使われる「コミ」を TUBSTAP のルールに導入することで，より多様なマップ作成が可能になるかもしれないと考えている．

参考文献

[1] 村山公志朗ら，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，GPW2013，pp 146–153

[2] 池田研究室： 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/

[3] 巡回セールスマン問題研究用ベンチマークデータセット TSPLIB，http://elib.zib.de/pub/Packages/mp-testdata/tsp/tsplib/tsplib.html

[4] Huang, S.-C., Muller, M.. Investigating the limits of Monte-Carlo tree search methods in computer Go. CG 2013,pp. 39–48

[5] 松原仁: コンピュータ将棋の進歩４　４章　次の一手形式によるコンピュータ将棋の評価，共立出版（2006）

[6] GAT (Game AI Tournaments @UEC)，〈http://minerva.cs.uec.ac.jp/gat_uec/wiki.cgi?page=\%C2\%E81\%B2\%F3GAT2016〉（参照 2016-09-28）

[7] ねおだ如: ゲームのアルゴリズム，ソフトバンククリエイティブ（2006）

[8] Bourg, D.M. and Seemann, G. :ゲーム開発者のための AI 入門，オライリー・ジャパン（2005）