

ターン制戦略ゲームにおける局面評価値構成のための局面分割および単純化ゲームのオフライン木探索

佐藤 直之^{1,a)} 藤木 翼^{1,b)} 池田 心^{1,c)}

概要: ターン制戦略ゲームは様々な商業タイトルが発売されており人気が高いジャンルであるが、AIプレイヤーの強さはまだ人間の上級者を超える水準に達していない。我々はターン制戦略ゲームにおける局面評価関数に着目し、駒のHP値の総和のみを考える従来の評価関数に代わって、駒の複雑な相性や位置関係も利用した評価関数を提案した。この手法はゲームの局面を少数の駒からなる複数の部分的な局面に分け、それぞれにゲームの単純化とオフラインな探索に基づいた評価値を与え、足し合わせる。我々はこの手法をターン制戦略ゲームプラットフォームのTUBSTAPを用いて実装した。提案手法を用いたAIはTUBSTAPの既存の強いAIに対してある条件下で66%の勝率を記録し、本手法の有効性が確認できた。

An approach to evaluate turn-based strategy game positions with offline tree searches in simplified games

NAOYUKI SATO^{1,a)} TSUBASA FUJIKI^{1,b)} KOKOLO IKEDA^{1,c)}

Abstract: Turn-based strategy games are a genre of games being popular but in which AI players are not competitive enough for advanced human players. We proposed a state evaluation function for the games utilizing information about unit locations and relationships about dis/advantageousness among units. At first, we assign evaluation values to partial game states by offline tree search with simplified game rules. Then, we sum up these values to estimate how favorable the whole game state is. We applied the method to TUBSTAP platform and our method overwhelmed DLNC-AAS AI player, which is the most competitive player in the TUBSTAP environment as far as we know, with 66% win rate in a certain setting.

1. はじめに

これまで古典的なボードゲームから新しいコンピュータゲームにわたり、沢山の種類のゲームのためにコンピュータプレイヤー（以下 AI と呼ぶ）が開発されてきた。それらの AI の思考ルーチンには様々な技法が使われるが、とりわけ一般的なのは木探索と局面評価関数の組み合わせで、チェスや将棋などの古典的な手番交代制ゲームに広く用いられた。

このような評価関数はゲーム局面に関係づけられた様々

な情報を利用する。チェスや将棋において自陣の駒の数や種類は人間にとって解りやすい情報であり、局面評価関数の一部によく利用される。同様にオセロの駒の数や五目並べの二連や三連の数などの情報も人間にとって扱いやすく、局面評価のよい指標となる。

とはいえ、より複雑な情報も考慮に含めたほうが局面評価の性能が向上するのが普通で、例えばオセロでは「開放度」という尺度が、将棋では2つの駒の位置関係といった情報がプログラム中の評価関数で広く用いられてプログラム全体の性能向上に貢献してきた [4]。これらの情報には機械学習などの手法により自動的に調整された重み係数がよく与えられる。

しかし局面評価関数の設計が難しいゲームもあり、例えば囲碁では石の生き死にの判別などが難しく、局面評価関

¹ 北陸先端科学技術大学院大学情報科学研究科
Japan Advanced Institute of Science and Technology, Information Science Department

a) satonao@jaist.ac.jp

b) s1310062@jaist.ac.jp

c) kokolo@jaist.ac.jp

数を用いずモンテカルロ木探索によるアプローチが一般的である。そして我々が着目するターン制戦略ゲームも高精度な局面評価関数の作成が難しい。

まず、ターン制戦略ゲームには駒の間に複雑な相性関係が設けられ、その有利不利が勝敗に大きな影響を持つ。特にその相性関係は3すくみな構造を含むのが普通で、それぞれの駒の価値は局面に応じて頻繁に変わる。また2駒の関係には先着して攻撃した側が著しく有利になるようなものもあるので、駒の距離関係の未来における推移も局面を大きく左右する。それらの特徴は駒に関する特徴量に静的な重み値を適切に設定する試みを困難にする。例えば藤木らの研究 [3] では特定のマップで人間に迫るレベルの AI が提案されているものの、そこで使われている局面評価関数は駒の HP の線形和のような単純なものである。

そこで我々は2人プレイヤーのターン制戦略ゲームの戦闘を対象として、HP の線形和よりも精度が高い局面評価関数の獲得を目指す。具体的には、ある局面の互いの駒の種類や位置関係の情報も用いて、適切に勝敗を予測できる局面評価関数を作りたい。

そのために、まずゲームの局面を少数な駒から成る部分的局面に（重複も許しながら）分け、それぞれに単純化ゲームの木探索に基づく評価値を割り当ててから足し合わせるアプローチを提案する。その部分局面のための単純化ゲームにおいて局面の情報やルールはある程度抽象化されているが、可能な行動は単なる攻撃だけでなく、駒の距離関係に変化を与える「接近」や「退避」といった行動もあって、双方のプレイヤーが最善を尽くした場合の勝敗をかなり高い精度で近似する。

こうして我々は駒の HP だけでなく位置関係や相性関係も考慮にいった、従来よりも精度の高い局面評価値の獲得を試みて、TUBSTAP プラットフォーム上にて既存の AI と対戦実験を行った。

2. 背景

2.1 ターン制戦略ゲーム

ターン制戦略ゲームあるいはターン制ストラテジーゲームは多数の駒を用いて複数のプレイヤーが争う形式のゲームである。有名なタイトルとして Civilization シリーズ [8] や Battle Of Wesnoth などがある、ビデオゲームである。

図 1 はターン制戦略ゲームのゲーム画面の例だが、これらのゲームは概して将棋やチェスと以下の点で大きく異なる。

- 1 ターンに複数の駒を動かせる。
- 盤上の駒は HP と呼ばれる数値を持ち、これがゼロになると盤から取り除かれる。
- 駒は隣接する敵駒への攻撃で HP を減らす。
- ゲームの初期盤面には様々な種類がある。マスには地形が設定され進入不能などの効果を持つ場合がある。

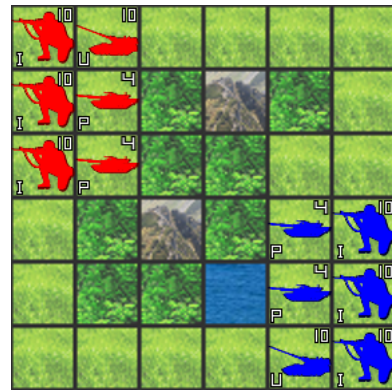


図 1 ターン制戦略ゲームの対戦局面例。『TUBSTAP』プラットフォーム。チェス等と異なり 1 ターンに複数の駒が動かせる。初期局面が多様で、様々な駒やマスの配置から勝負が開始される。

またゲームのタイトルによっては、占領や生産といった自陣の勢力に変動を与える戦略行動や、「偵察」を行うまで敵の駒の位置が解らない不完全情報性など、より複雑なルールが組み込まれている。しかしそういった複雑なルールを無視したとしても、ターン制戦略ゲームで強い AI の作成は難しい。

まず、上に列挙したような特徴のために探索空間が非常に大きく、例えば自陣に平均合法手数 が 10 の駒が 6 体いる場合は 1 ターンにプレイヤーがとれる行動の数は $10^6 \times 6!$ 通りにもなる。そしてこれより平均合法手数や駒の数が多い状況は実際のゲームで珍しくない。また駒の種類や相性のため、ある駒は特定の駒に大きなダメージを与えられたり、逆にまったく与えられなかったりする。そのせいでたった 1 つの駒が敵の駒 5 個を相手にして勝ち、すべて盤から取り除いてしまう事も起こり得る。

そのためターン制戦略ゲームの（上述の特殊な行動を省いた）シンプルな戦闘だけに問題を限定しても人間プレイヤーより強い AI を作る事は難しく、我々はこうした戦闘の状況を対象問題とする。

2.2 関連研究

我々は大きく複雑なゲーム局面の評価を、少数の駒に着目した部分的局面の評価値の和により近似している。そしてそれら部分的局面の評価値はオフラインで計算しておきテーブルとして利用する。こうしたアプローチは将棋プログラムの Bonanza [4] 等でも、全体の局面評価値を少数の駒の位置関係からなる単位的な特徴量の足しあわせで構成する形でしばしばみられる。また、チェッカーの求解 [5] に用いられた終盤のデータベースやチェスのエンドゲームデータテーブル [6] も問題を分割してオフライン計算しておきテーブルとして利用する点では我々のアプローチと類似している。また我々のアプローチとさらに近い試みがトランプの 1 人ゲームであるカルキュレーションで行われている [7]。その研究ではゲーム中の局面を部分的局面に分

けて、単純化ゲームで解析し、個々の成功率の積によって全体の局面の成功率を見積もっており、実装 AI が人間の上級者を超える性能を示した。

ターン制戦略ゲームのプラットフォームについては、商業タイトルの Civilization[8] をモデルにした Freeciv[9] があり、多くの研究に利用されている。Freeciv は内政や外交といった複雑な要素も含んでいて、資源の活用や都市開発などの最適化に関する研究 [10][11] が試みられている。

さらに、ターン制戦略ゲームの駒同士の移動と攻撃による戦闘のみに着目したプラットフォームとしては TUBSTAP[1] があり、これは Freeciv よりも複雑な要素が少なく、複数の駒の単純な戦闘に焦点が当てられている。このプラットフォーム上ではモンテカルロ木探索のアプローチで UCT ベースの AI やシミュレーション深さを限定して局面評価関数を組み合わせる手法 [3] が提案された。他にターン制戦略ゲームの戦闘に着目した研究には、市販タイトルをモデルに独自に作成したゲーム上で、不要と予想される移動行動の枝刈りを伴う UCT 探索を行うものがある [2]。

さて、本稿で我々はターン制戦略ゲームの一例として TUBSTAP プラットフォームを環境として選んだ。その詳細を次項で説明する。

2.3 TUBSTAP

2.1. で述べたターン制戦略ゲームの持つ基本的な要素に加え、TUBSTAP は具体的に以下のルールを持つ。

- 駒：戦闘機 (F)、攻撃機 (A)、戦車 (P)、対空戦車 (R)、歩兵 (I)、自走砲 (U) の 6 種類の駒があり、これらの駒は図 2 のような相性を持ち、これが攻撃のダメージ量に影響を与える。
- HP：駒の HP は 1 以上から 10 以下に設定される。
- 地形：ゲームの局面 (マップ) の駒が通過できるマスにはそれぞれ道路、平地、陣地、林、山、海のいずれかの地形が割り当てられており、その位置にいる駒が受けるダメージを軽減させたり、また通過する特定の駒の移動範囲に制限を与える。
- 勝利条件：将棋やチェスと違い、相手の駒の全滅により勝敗が決まる。ただしターン数に上限が設けられている場合はそのターン数を超過した場合に判定 (駒の HP 残量の総和による) によって勝敗が決まる。

この環境は既存のターン制戦略ゲームから内政や占領、キャラクターの成長要素などが排除され複数着手性のみが強調された 2 人ゼロ和有限確定完全情報ゲームである。そのため既存のターン制戦略ゲームよりシンプルであるが将棋などのゲームと比べて複雑であるといえる。

3. 提案手法

ターン制戦略ゲームの戦闘において、多様な相性関係に

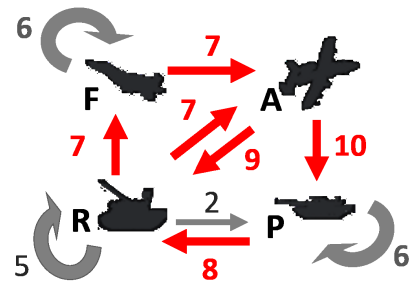


図2 TUBSTAPの駒の相性。数字は攻撃側の駒がHP10のときその相手に与えるダメージを示す。赤い矢印の先の駒に相性が良く、相互に赤い矢印が向いている2駒は互いが互いに大きなダメージを与えるため先着した側が有利となる。

あるそれぞれの駒が、相手に近づいたり、攻撃を加えたり、あるいは一時的に逃げたりしながらゲームを自軍の有利になるように進めようとする。それらの行動が生み出す状態遷移の可能性は多様で、残存する駒のHPや種類に基づく単なる重み付けのアプローチでその将来の勝敗を正確に近似するのは難しいと思われる。そこで我々は大きな局面を少数の駒からの局面に分け、それぞれの勝敗を単純化ゲーム上のMin-Max探索で予測してそれらを評価値に反映するアプローチを提案する。

そのために我々の手法はまずゲームの部分的な局面における理論的な勝敗を、ゲームの単純化とMin-Max木探索によってオフラインで近似的に計算しておく。そしてあるゲーム局面において、少数の駒からのいくつかの部分的局面を抜き出し、それぞれの近似された勝敗または有利さを反映する評価値の足しあわせを行って元のゲーム局面の評価値に利用する。

この提案手法はあくまで局面評価関数の生成のみに着目したものであり、実際のAIへの使用にあたってはなんらかの木探索手法と組み合わせる必要がある。

3.1 部分局面の抜き出しと単純化

まず本手法ではTUBSTAPのゲーム局面を両プレイヤーの駒3つずつ(計6駒)からなる部分的な局面に着目することにする。例えば図3では12駒からなるゲーム局面から赤の戦車、攻撃機、歩兵、青の戦闘機、戦車、攻撃機、からなる部分局面を抜き出した。そして手番のプレイヤーから見て敵の駒それぞれに攻撃が届くかどうかを調べ、それを2つの駒の間の枝として表現した無向グラフを生成する。

このグラフは手番プレイヤーと駒の種類とHP、そして2駒の距離が攻撃射程の内か外かを情報として持つ。ちなみに攻撃射程に関する情報は対称でない(片方の駒から相手へ攻撃が届くが逆には届かない場合がある)が、本手法ではこれを無向な枝として扱うことにする。また手番側の駒の一部のものだけが行動済みであるような局面も考慮せず、あるプレイヤーの全ての駒が行動する前か、行動を終えた後

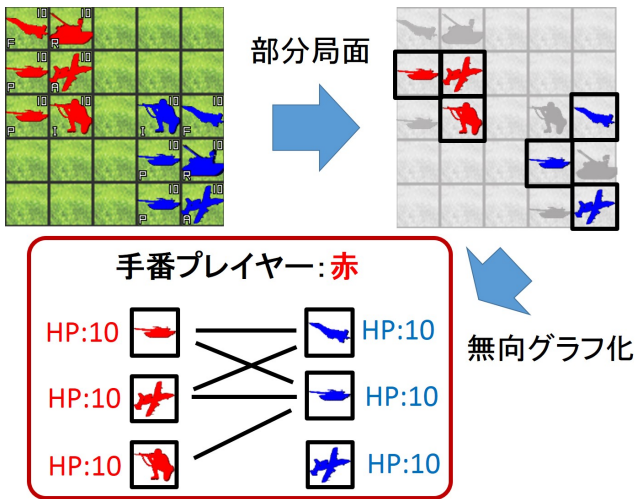


図3 部分局面抜き出し例. 特定の駒に注目し, 無向グラフを作る. 手番側の攻撃が届く場合に2駒間に枝を作る.

の局面のみ評価の対象としている. こうした事情は計算コスト等の事情による. こうして作られたグラフが, 単純化されたゲームにおける1つの状態になる.

3.2 単純化されたゲーム局面に対する評価値計算

各無向グラフの好ましさは, 行動選択により状態を遷移させて勝敗の計算を行うことで得ることとする. 単純化されたゲームの中では, 以下の4種類の行動が可能である.

- 攻撃: 枝で接続された駒1つに攻撃を加える. その後, 攻撃を実行した駒と相手の全ての駒との間に枝をつなげる.
- 接近: 実行した駒と相手の全ての駒との間に枝をつなげる.
- 退避: 実行した駒と相手の全ての駒との間の枝を消す. 駒は, 2回連続しての退避は選択できない.
- 無行動: 何もしない. HPが0の駒はこの行動しか選択できない.

書く手番では3つの駒それぞれがいずれかの行動を実行する. 図4にプレイヤーの行動選択と局面の移り変わりの例を示す. ちなみに本稿の実装ではある手番中の駒の行動順は一定で, 順序の入れ替えによる結果の変化は考慮していない. そしてどちらかの駒のHPが全て0になるか, 互いにダメージを与えられない状況になれば終端である.

攻撃行動の後に敵の駒すべてと枝がつながる仕様は, 実際のゲームでは相手の駒に接近して攻撃を加えることで相手の他の駒の攻撃射程に入ってしまう現象が多く起こるため設定した. また, 退避行動が連続して2回選べないのはあきらかに不利な状況にある側がずっと退避を選んで局面を引き分けに持ち込めないようにするためである. そして実際のゲームでも有限な広さのマップにおいて敵の駒から永久には逃げ続けられない事が多い.

末端局面の評価値は互いの駒のHPの総和の差とし, 充

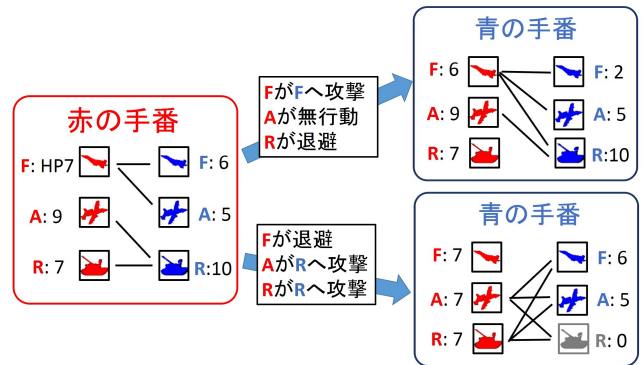


図4 単純化ゲーム内の行動選択による状態遷移. どちらかの陣営の駒のHPが全て0になれば終端状態で, 敵と味方の駒の合計HPの差を評価値として返す.

分な深さ(ゲーム終端まで読み切り)のMin-Max型探索によってルート局面の評価値が得られる. 評価値が正ならルート局面の手番お勝ちで, 負なら負けだが, その値の大きさを見る事により局面の大勝, 大敗, 僅差の辛勝などを区別できる.

3.3 単純化されたゲームの局面勝敗値のオフライン計算および活用

こうして単純化ゲームの局面評価値を, ゲーム中で生じうる全局面に対してオフラインに計算しておき, テーブルの形で利用することにする. 具体的には図5のように, ゲーム中でその価値を判断したい局面からいくつかの3駒同士の部分局面を取り出し, 各部分局面に対応する評価値をテーブルから取得する. そしてそれらの和を元の局面の評価値の主たる成分として用いる.

この部分局面の取り出し方にはいろいろなやり方があり得て, 駒の重複や, または逆に一回も部分局面の要素に選ばれない駒がある事も許容される. どのような取り出し方が適切なのかは局面や組み合わせる探索手法によっても異なり, その選択指針を提示することはまだできない. そのため本稿における部分的局面の取り出し方は, 駒の情報をなるべく多く抜き出しながらも計算コストが増えすぎないような, ある恣意的な選択による.

3.4 手法の特性

3.4.1 期待できる特長

本手法が持つと思われる特長を述べる.

- オンライン計算時間の抑制
単純化ゲームの評価値をオフラインで計算しておきテーブルで利用するため高速である. 1つの局面に評価値を適用する際の計算量は, そこから取り出す部分局面の数に対する線型のオーダーで抑えられる.
- 部分局面に対する勝敗近似の精度の高さ
単純化ゲームの中では各駒の種類とHPと, 射程距離に関する位置情報が利用される. そしてその上で攻撃

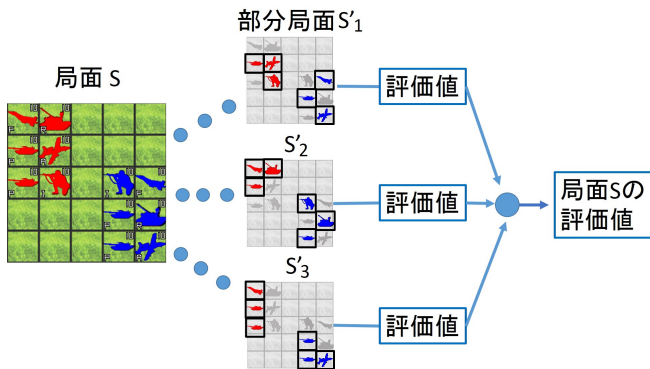


図5 局面への評価値計算. いくつかの部分局面を取り出し、オフライン計算の結果をそれぞれに与えて合計する. 部分局面の取り出し方に特に制限はなく、設計者が自由に決められる.

行動や、射程距離への出入りの行動を網羅してオフラインな木探索が行われる. よって部分局面の実際の勝敗をかなり高い精度で近似することが期待できる.

特に後者の特長については、駒それぞれの、他の駒との(図2に示されるような)複雑な相性関係の中で変動する価値などを手動の特微量設計と重み値決定によらず適切に定めることができ、あるいは一般的な機械学習で必要とされるような教師データも必要としない. また、シミュレーション手法と比べてオンラインでの計算コストが軽く、あるいはシミュレーションを途中で打ち切ってその局面を評価する使い方もできる.

3.4.2 解決すべき問題点

一方で、提案手法の適用にあたって対処が必要な点も多く、今後の研究が必要である.

- メモリサイズ

本手法で用いる単純化ゲームの評価値テーブルはサイズが大きい. ある3駒同士からなる単純化ゲームについて各駒のHPを0以上10以下としてテーブルを作ろうとすると、枝のパターンが $2^9 (= 512)$ 通りでHPのパターンが $11^6 (= 1,771,561)$ 通りなので907,039,232個の評価値をメモリに格納する必要が出てくる.

- オフラインの計算時間

また、オフラインでの計算時間の大きさも課題の一つで、ある1つの(敵味方3駒ずつの)評価値テーブル作成に際して20コアの高速計算サーバで約4日間の時間を必要とした. こうしたテーブルを将来的にあらゆる駒種類6個の組み合わせについて作ろうとすると、オフラインとはいえかなり長い計算時間がテーブル作成にかかることになる.

- 遠距離ユニットと地形効果

本手法は移動と攻撃を同時にできない代わりに攻撃射程が2以上あるような、いわゆる『遠距離攻撃ユニット』の存在を考慮していない. また本手法は現在、地形効果の影響を考慮に入れていない. しかし遠距離攻

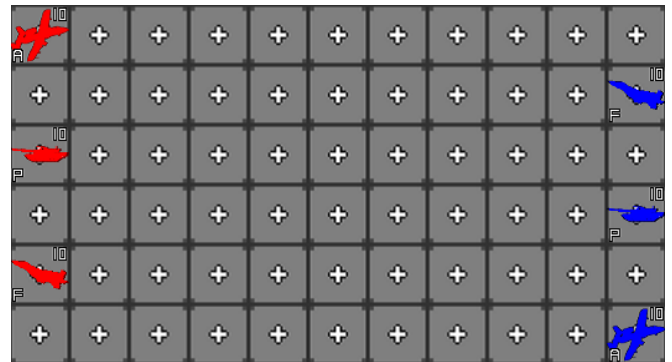


図6 実験に用いるマップの1つ. 地形は全て「道路」で、遠距離ユニットを含まず、最初のターンに攻撃が届かない程度に互いの駒たちが離れている.

撃ユニットは移動と攻撃の応酬について他のユニットとかなり異なる様式を持ち、また地形効果もユニットの生死に無視できない影響力を持つ. よってそれらの要素が含まれるマップでは提案手法の性能が落ちる事が予想される.

しかしこれらの問題点について、メモリやオフライン計算時間の問題は入力や出力の数値を適度にグルーピングする事で軽減できると予想でき、また地形や遠距離攻撃ユニットについてもゲームの簡略化についてある程度の拡張をほどこし対処が可能と考えている.

4. 性能調査実験1: 局面分割無し+全幅探索

なるべく単純な場合から手法の性能を調査するため、まず我々は6駒の開始局面となるマップで対戦実験を行った. これらのマップで部分局面の取り出し方は1通りしかなく、また全部の着手も1手分の深さまでなら現実的な時間で読みきれる.

4.1 使用マップ

対戦実験に使用するマップは4種類用意し、それぞれ図6のようなサイズと駒配置である. 全てのマップで両プレイヤーは駒の種類と配置が対称だが、それぞれのマップで使用する駒の種類は、戦闘機(F)、攻撃機(A)、戦車(P)、対空戦車(R)のうち{F, A, P}, {F, A, R}, {F, P, R}, そして{A, R, P}, の4通りとなる. 駒の初期HPは全て10とした. それぞれのマップで対戦は400戦行われ、200戦ごとに先手番と後手番が交換される. 開始から13ターンで勝敗判定が起り、HP合計が高い側が勝ちとなる. ただし合計が等しい場合に引き分けとなり、その試合は勝率計算において0.5勝と勘定される.

4.2 提案手法 AI

対戦に用いるAIとして、全幅探索と提案手法による局面評価関数を組み合わせるAIを用意した. 深さ1(手番開始局面から自陣の駒が全て行動を選択し終えるま

度を深さ 1 と想定する) の全幅探索と提案手法による局面評価関数を用いる。局面評価関数は具体的に以下の形で表される。

$$E(s) = \text{Bias}(s) - c_d \cdot \frac{1}{2n(n-1)} \sum_{1 \leq i < j \leq n} d_1(u_i, u_j) + c_{HP} \cdot \{\sum hp(u_{friend}) - \sum hp(u_{enemy})\} \quad (1)$$

ただし $\text{Bias}(s)$ は提案手法の項で、 s の単純化された部分局面へのオフライン評価値の和である。右辺第二稿は盤の全ての 2 駒の距離の平均値であり d_1 は 2 つの駒の距離 (マンハッタン距離とする) を返す関数で n はマップの駒の総数である。第三項は各陣営の駒の HP の差を表し、 c_d と c_{HP} はそれぞれ各項のバランスを取るための定数である。

各駒の距離を評価値からマイナスする理由は、各駒間の距離が近いと局面の遷移が単純化ゲームの内容に一致しやすくなるためである (例えば味方の駒が密集していると、攻撃してきた敵の駒は味方の駒全ての攻撃射程内にきちんと入る)。ただし c_d と c_{HP} は 1 よりかなり小さく、それらの項はバイアスの項の値が同一な局面間でのみ影響を持つ。そして c_d は c_{HP} の $\frac{1}{20}$ 倍であり、平均距離より HP 値の項が優先度は高い。

4.3 比較 AI

我々は、比較対象の AI として以下の 2 種を選んだ。

- UCT: モンテカルロ木探索の一種である UCT 法を用いて行動評価を行う [12]。1 ターンで行うプレイアウト回数は (6000 × 自陣の駒の数) であり、ノードの勝率と UCB 項をバランスする係数は 1.0 である。なお次に述べる DLMC+攻撃行動探索や提案手法 AI と異なり、自陣の駒を 1 つ動かすたびに新たに探索を行っている。
- DLMC+攻撃行動探索: シミュレーションを一定の深さで打ち切り、局面評価関数を用いるモンテカルロ木探索により行動評価を行う [12]。それぞれのターンで、全部の駒の行動組み合わせのうちランダムに 500 個と、それぞれのサンプルで 100 回の深さ 3 シミュレーションを行う。そして、可能な攻撃行動の 3 つの組み合わせ全てのうち、実行後の局面がもっとも高い評価値 (シミュレーション後の局面に用いるものと同じ評価関数による) を持つ組み合わせ 1 つを選び、この攻撃行動 3 つを適用したあと同様に深さ 3 シミュレーションを行う。シミュレーションの末端で用いる局面評価関数は自分と相手の駒の HP の総和の差を返す。

前者 (UCT) は、あるシンプルな手法デザインを持つ相手として、後者 (DLMC+攻撃行動探索) は TUBSTAP の既存手法で最も強いものとして、それぞれ比較対象に選んだ。これらのパラメータは思考時間がだいたい同じになる

表 1 各 3 対 3 マップにおける提案手法 AI の勝率

マップ	{F, A, P}	{F, A, R}	{F, P, R}	{A, R, P}
対 UCT	65.6%	93.0	76.5	89.8
対 DLMC+攻撃行動探索	61.5%	74.5	64.6	69.3

ように設定され、WindowsOS の 3.4GHz、8GB メモリ、8 スレッドのマシンで 1 ターンあたり約 10 秒以内の思考時間になる。

4.4 結果

結果を表 1 に示す。全ての場合で提案手法が勝ち越している。特に現在我々が TUBSTAP 環境で得られる最も強い AI の、深さ限定モンテカルロ法と攻撃行動探索の組み合わせ AI も有意に性能で上回っている。つまり提案手法が 3 対 3 の部分的局面に与える評価値はかなり精度が高いと考えられる。

5. 性能調査実験 2 : 局面分割あり + 枝刈りを伴う全幅探索

先ほどより少し複雑な状況設定として、部分的局面の足しあわせを行う局面のマップについて実験を設けた。互いのプレイヤーは 4 駒ずつで勝負を開始するため、3 駒ずつの局面用の評価値がそのままでは使えず、それらの足しあわせなどを検討しなくてはならない。また行動全てを考慮する全幅探索では計算時間が長くなるため、ある種の枝刈りを行う必要があるのも先の実験とは異なる。

5.1 使用マップ

マップを 2 種類用意し、それらは両陣営に戦闘機 (F)、攻撃機 (A)、戦車 (P)、対空戦車 (R) の駒を 1 つずつ含む。駒の初期 HP は全て 10 で、サイズは縦 11 横 8 の広いものと縦 5 横 5 の狭いものがあり、広い方では 1 ターン目に先手プレイヤーは敵の駒に攻撃が届かず、狭い方では 1 ターン目からそれぞれの駒は全ての敵駒に攻撃が届く。また狭い方では先手の F,A,P,R の HP だけがそれぞれ 7,8,8,8 となっており最善手の発見が簡単になりすぎないような工夫を施した。対戦は各 400 戦行われ、全体の 200 戦ごとに先手番と後手番が交換される。開始から 13 ターンで勝敗判定が起こり、HP 合計が高い側が勝ちとなる。ただし合計が等しい場合に引き分けとなり、その試合は勝率計算において 0.5 勝と勘定される。

5.2 使用 AI

そして対戦に用いる AI は前項の実験と同じ 3 つである。提案手法の評価関数は前項の式 (1) と同一である。今回の実験では複数の部分局面の取り出しと評価値の合計が必要になるが、部分局面は図 7 のように互いのプレイヤーの 4 つの駒のうち同じ 3 駒ずつを取り出して作る。

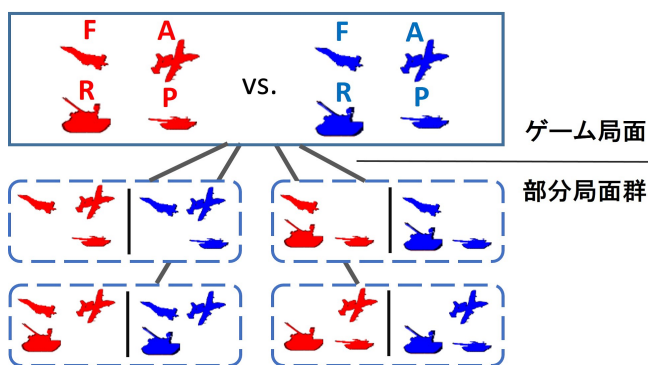


図7 4対4のマップに対して我々が行った、部分局面の取り出し方。このような6駒からの部分局面4つを取り出し、それぞれの評価値を合計して元の局面の評価値を作った。

表2 各4対4マップにおける提案手法 AI の勝率

マップ	{F, A, P, R} - Large	{F, A, P, R} - Small
対 UCT	79.0%	71.9
対 DLMC+攻撃行動探索	61.8%	65.3

部分局面の取り出し方のうち最も細かいものを想定すれば、先手と後手の駒3つずつを取り出して $400(=6C_3 \times 6C_3)$ 通り考えられる。そうしたパターンのうち例えば先手の {F, A, R} 対後手の {A, R, P} のような、駒種類が非対称な部分局面に対する計算がオフラインの計算コストの事情で省かれている。

また提案手法は思考時間を縮小するために探索で駒の移動行動の半分を無作為に枝刈りしていて、4つの駒が操作できる状況で約16分の1に移動行動の数を縮小できる。このような移動行動の枝刈りを提案手法 AI だけに適用しているが、UCT や DLMC+攻撃行動探索のような複数の行動からいくつかだけサンプリングしてくる設計の AI にはこうした措置は不要である。

5.3 結果

結果を表2に示す。全てのマップで提案手法による評価関数と半分の枝刈りを伴う深さ1全幅探索が他のモンテカルロ系手法に有意に勝ち越している。よって駒数の少ない小規模な局面で部分局面の足しあわせを行う場合に我々の手法は高い性能を発揮したといえる。

6. 性能調査実験3：局面分割あり+行動の無作為なサンプリングによる木探索

前項までの実験で使用したマップは両プレイヤーの陣営が3駒または4駒ずつで、この程度の駒数の少なさのマップは実際のターン制戦略ゲームで一般的とはいえない。そこで両陣営が6駒ずつからなる12駒のマップで対戦実験を行った。

そしてこの程度の駒数のマップとなると探索空間が広く、前節までの全幅探索型の手法では適切な思考時間で着

表3 従来型の局面評価関数を用いる AI に対する提案手法評価関数 AI の勝率 [6対6のマップ]

マップ	F-A-P	F-A-R	F-P-R	A-P-R
対 DLMC+AAAS [既存局面評価関数]	58.1%	44.3%	53.5%	66.5%

手生成がしにくい。そこで我々は、深さ限定モンテカルロ木探索 AI の局面評価関数の部分だけを提案手法に置き換えて既存の評価関数との性能の差分を見る事にした。つまり他のモンテカルロ木探索ベースの AI 同様に可能な着手をランダムにサンプルして局面に適用し、シミュレーション後の局面に式(1)の局面評価関数を用いている。

6.1 使用マップ

縦11横11マスのマップを4種用意した。それぞれのマップで両陣営の駒配置は対称で、{F, F, A, A, P, P}, {F, F, A, A, R, R}, {F, F, P, P, R, R}, または {A, A, P, P, R, R} の6駒ずつである。それらがマップの対角上に配置され、1ターン目に先手プレイヤーの駒は相手プレイヤーの駒に攻撃が届かない。引き分けの扱い等は先述の実験と同等である。

6.2 使用 AI

対戦に使うのは前項と同様の DLMC+攻撃行動探索型 AI であり、その局面評価関数を提案手法(式1)とするか従来通りの「HP値総和の差」とするかで、2つの AI (提案手法 AI と比較 AI) を用意した。評価関数だけの違いに焦点をあてているため、この実験ではUCT法による AI は用いていない。

提案手法はこれらのマップで局面を64個の部分局面に分ける。例えば図8のように両プレイヤーが {F, F, A, A, P, P} を持つマップでは互いの {F, A, P} 3体ずつの構造に注目し、 $2^6(=64)$ 通りの部分局面が取り出せる。第4項の実験と比べて約64倍の計算が必要になっているが、単なるテーブルの参照が複数回行われるだけであり、全体の着手生成の中で計算量時間は数%ほどしか増大しない。

末端局面での局面評価関数を「HP値総和の差」とする既存手法 AI には(行動サンプル数, シミュレーション回数)のパラメータとして(250, 50)を与え、局面評価関数を提案手法により行う AI はそれらのパラメータを(240, 48)とした。そのパラメータ設定で両 AI の思考時間がほぼ一致する。

6.3 結果

結果を表3に示す。表記が煩雑になるのを防ぐため、{F, F, A, A, P, P} の両プレイヤー6駒ずつ計12駒のマップを単に F-A-P のマップと記した。

この4種のマップで95%信頼区間は±5%より狭く、F-P-R のマップ以外では提案手法は有意に強いが有意に弱い

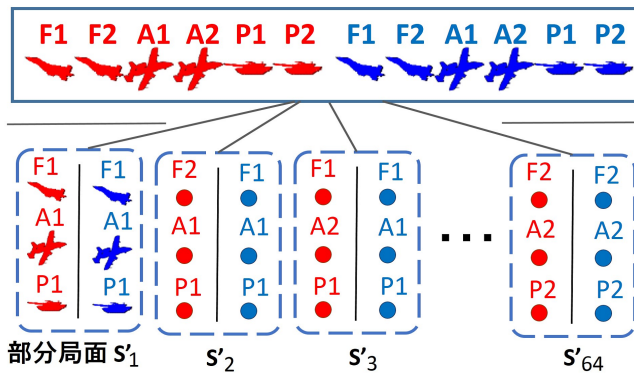


図8 本実験の12駒のマップで行った部分局面の取り方。1つの局面から互いのF, A, P同士からなる部分局面を $2^6 (= 64)$ 通り取り出している。

という結果になった。2つのマップで提案手法は有意に高い勝率を導くが、1つのマップでは勝率が50%より低くなる。その原因には主に、一定深さのランダムシミュレーションによって駒ごとの距離関係の推移が現実的なものからズレて、提案手法の局面評価に悪い影響が及ぼされた事が考えられている。しかしこの問題には、ランダムシミュレーション適用後の局面を評価する目的のための、局面の距離関係の扱い方を単純にした改造をほどこした評価関数の設計により対処できると想定できる。また探索手法と局面評価関数のマッチングの問題や、あるいは局面評価の際の部分局面の取り出し方および各評価値の組み合わせ方の問題も検討の価値がある。

とはいえ、本実験における提案手法による局面評価関数の用い方はかなりナイーブで、それでも2つのマップで有意に性能を改善させており、本手法はある程度の拡張で性能をさらに向上させる余地があると考えている。

7. まとめ

本稿ではターン制戦略ゲームにおいて駒の複雑な相性や位置関係を反映した局面評価関数を作ることを試みた。そのためにゲームを複数の単純化された部分的な局面に分け、オフラインで解析した評価値を与え、それらを合算することで全体の局面の評価を行う手法を提案した。そして提案手法の実装をTUBSTAP環境を用いて行った。

単一な部分局面によるマップで提案手法が精度の高い評価値を生成する事を確認した後、我々は8個および12個の駒からなるマップを6種類用いて提案手法と既存の局面評価関数の性能差を観測した。その結果、1つのマップで勝率が劣ったものの、ほとんどのマップで有意に高い勝率を記録する事を確認できた。

ただし今回の実装にはさまざまな面で検討が不十分な点があり、将来の課題も多い。まず部分的局面の取り出し方と部分的な評価値の合算について多くの選択肢があるため、そのうちの有効な方法を調べる必要がある。特に現在

我々は小さい部分局面で精度の良い評価値を生成してからそれを合算しているが、それよりも、合算したときに全体的な局面に（結果的に）精度の高い評価値を与えるように手法のデザインを再考する事は重要である。

6章で行った実験のように何手かのランダムシミュレーション後の局面を本手法で評価するためには、位置に関する情報にどれほど信頼性を置くかは微妙で、駒のHPと相性に重きを置いた方が評価の精度が良くなる可能性がある。つまり具体的に、攻撃射程の内外に関する情報を省略しつつ、より多様なパターンの部分局面の取り出し（例えば先手の{P, P, A}対後手の{F, A, P}のような非対称な構造）を行うアプローチが考えられる。

またメモリサイズの問題を克服するための拡張も重要である。具体的には出力評価値と入力HP値に関するグルーピングで各テーブルのサイズをおさえるアプローチを考えている。

参考文献

- [1] 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/ikedalab/tbs> (2015/9/17).
- [2] 加藤, 三輪, 鶴岡, 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. IPSJ-GPW 2013, pp.138-145.
- [3] 藤木, 村山, 池田. 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, 第8回E&Cシンポジウム, 2014-3-19.
- [4] K. Hoki and T. Kaneko. Large-Scale Optimization for Evaluation Functions with Minimax Search, *Journal of Artificial Intelligence Research*, 49: 527-568, 2014.
- [5] J. Schaeffer, et al. Checkers is solved. *science*, 317.5844: 1518-1522, 2007.
- [6] H. Ernst A. Endgame databases and efficient index schemes for chess, *International Computer-Chess Association journal*, 22(1): 22-32, 1999.
- [7] 田中. 経験則を用いないカルキュレーションのプレイ, IPSJ-GPW 1999, pp.76-83, 1999.
- [8] Sid Meier's Civilization V, <http://www.civilization5.com/> (2015/9/17).
- [9] The Freeciv Wiki, http://freeciv.wikia.com/wiki/Main_Page/ (2015/9/17).
- [10] T. R. Hinrichs and K. D. Forbus. Analogical Learning in a Turn-Based Strategy Game, *Proceedings of the 21st International Joint Conference on Artificial Intelligence*, pp. 853-858, 2007.
- [11] S. Wender and I. Watson. Using reinforcement learning for city site selection in the turn-based strategy game Civilization IV, the 2008 IEEE Symposium on Computational Intelligence and Games, pp. 372-377, 2008.
- [12] 藤木, 村山, 池田. 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, IPSJ-GPW 2014, pp. 32-39.