

# 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

加藤 千裕<sup>1</sup> 三輪 誠<sup>2</sup> 鶴岡 慶雅<sup>3</sup> 近山 隆<sup>3</sup>

概要：ターン制ストラテジーゲームは、囲碁や将棋と比べて1ターンの候補手があまりにも膨大である。この性質が、ターン制ストラテジーゲームの分野において強いAIを作る妨げとなっている。しかしながら、一方で、ターン制ストラテジーゲームには戦術レベルでは盤面の局所性や、候補手の類似性が存在し、これによって探索空間を減らすことが可能である。そこで、本研究では、コンピュータによる問題の枝刈り、それによる探索空間の縮小を考える。探索にはゲームの分野で成果を上げている Upper Confidence Bound applied to Trees を用い、選択枝を枝刈りし、本研究用に作成したゲームを用いて、探索の効率化と強さの評価を行う。結果として、単純なUCTに対して良い成績を収め、枝刈りが有効であることを確認できた。

## UCT and Its Enhancement for Tactical Decisions in Turn-based Strategy Games

CHIHIRO KATO<sup>1</sup> MAKOTO MIWA<sup>2</sup> YOSHIMASA TSURUOKA<sup>3</sup> TAKASHI CHIKAYAMA<sup>3</sup>

**Abstract:** The number of possible actions at each turn in turn-based strategy games can be prohibitively large compared to that of Go and Shogi, and this makes it hard to build strong AI engines for such strategy games. However, the possible actions in turn-based strategy games usually have at least two characteristics that one could take advantage of in order to reduce the branching factor: locality of each battle field and similarity between actions. In this work, we propose a simple pruning method for reducing the search space. We implement the pruning method in the UCT algorithm and evaluate its effectiveness in terms of playing strength and computational cost, using a simple turn-based strategy game that we have developed for a research purpose. Our experimental results show that the proposed pruning method has improved the performance of the UCT search.

### 1. はじめに

ターン制ストラテジーは、多数の駒を操作して戦闘を行う、二人以上の多人数を前提としたシミュレーションゲームの一分野である。既存のゲームとしては Civilization [2], タクティクスオウガ [1], Battle of Wesnoth [3] といったゲームが該当するが、ゲームごとに要素は様々である。例えば有名なオープンソースの Civilization というゲームは、

自らの勢力を拡大していくゲームで、外交や同盟といった長期的な戦略が重要なファクターを持つ。一方タクティクスオウガや、Battle of Wesnoth に代表されるように、戦闘のみに重点を置いたゲームも存在する。多くのターン制ストラテジーは、プレイヤーにすべての情報が開示されているが、不完全情報のものも少なくない。例えば、Battle of Wesnoth は偵察を行うまで敵の居場所が分からない。このようなターン制ストラテジーゲームは一人で遊ぶことも多く、ある程度強いAIの存在はゲームを楽しむために重要である。しかし、既存のコンピュータプレイヤーには問題点が多い。例えば、Civilization のようなマクロな戦略を重視するゲームにおいては、ルールベースでのAI作成は非常に難しく、先読みをしようにも膨大な探索空間を必要と

<sup>1</sup> 東京大学工学部電子情報工学科  
School of Electrical and Electronic Engineering, Faculty of Engineering, The University of Tokyo  
<sup>2</sup> マンチェスター大学コンピュータ科学科  
School of Computer Science, The University of Manchester  
<sup>3</sup> 東京大学大学院工学系研究科  
Graduate School of Engineering, The University of Tokyo

する。また、戦闘にのみ重点を置いた、長期的な戦略を必要としないようなゲームであっても、囲碁や将棋と比べて盤面が広く、パラメータも各駒に対して複数定義されていることがほとんどで、探索空間は指数関数的に増大する。例えば、移動と攻撃をすべて含めて 50 通りの行動ができる駒が 10 体いるとすると、1 ターンに全ての駒を動かす場合、1 ターン内の行動の種類は  $10^{16} (= 50^{10} \times 10!)$  のオーダーまで広がってしまう。ここから更に相手の動きまで先読みしようとすると、膨大な空間を探索する必要がある。そこで、合法手を枝刈りして纏めるというのは、有効な手段であると思われる。

そこで本論文では、ターン制ストラテジーゲームを対象に、候補手に対し、枝刈りを行う手法を提案する。この手法により、選択枝を減らし、より深い探索を行って最適に近い解を見つけることができると考えられる。また、選択枝が少なくなることで、無駄な選択枝をプレイアウトで選ばなくなることによる探索の効率化も見込まれる。探索には Upper Confidence Bound applied to Trees (UCT) [4] を用い、評価としては候補手を抽象化したことによる強さと探索効率の評価を用いる。

本論文では、まず、2 章で本研究で利用する UCT と、対象にしたターン制ストラテジーについての関連研究を示す。次に 3 章で評価に用いたターン制ストラテジーゲームの概要を説明し、4 章で提案手法について述べる。そして 5 章で評価を行い、6 章でそれに対する考察を行う。最後に 7 章で本研究のまとめ、今後の計画について述べる。

## 2. 関連研究

### 2.1 モンテカルロ木探索

モンテカルロ法は、候補手の中で有望なものを、ランダムなシミュレーションを繰り返すことで選択するアルゴリズムである [5]。

原始モンテカルロ法と呼ばれる手法は、モンテカルロ法の中で最も単純なものである。候補手をひとつ選択して、その結果を得ることをプレイアウトと呼ぶ。それぞれの候補手について、ランダムに終局までシミュレーションをして結果を反映することを繰り返し、一番勝率の高かったものを選ぶ。しかしこのやり方では、相手のミスに期待するような手を選ぶことが多く、かきこい AI とは言い難かった。この原因として、有望な手を選んで深く探索しないため、相手が有望な手を選ばないことに期待してしまうということが挙げられる。

モンテカルロ木探索 [5] は、原始モンテカルロ法を改良したものである。この方法では、より有望な手、またはまだプレイアウト回数が少なく、調べる価値のある手に多くのプレイアウトを割り当てる。そして、ある候補手のプレイアウト回数が閾値を越えた場合、木を展開してプレイアウトを開始する接点をひとつ深くする。このため、プレイ

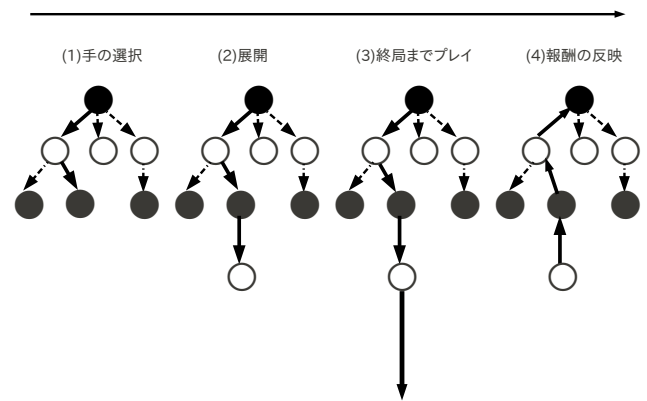


図 1 根ノードにおけるプレイアウト

アウトを繰り返すことでより良い手をより深く探索することが出来る。先ほど問題点として挙げた、相手のミスに期待するという問題も、相手の手番まで木を展開して探索を行うことによって解消される。モンテカルロ木探索の利点として、局面ごとの報酬を見る必要がなく、ぱっと見て戦況が把握し辛いターン制ストラテジーゲームに向いている。また、途中で探索を止めても最適解に近いものを出力してくれるため、探索時間が限られている場合にも有効な手法となりうる。

モンテカルロ木探索の具体的な流れは以下ようになる。

1. ルート局面から、有望な子ノード（次の局面）またはまだ十分調べていない子ノードを選択しながら枝（候補手）をたどり、末端まで行く
2. 末端のノードのプレイアウト回数が閾値を越えていた場合、もう一段展開し、子ノードを 1 と同様の基準で選択する
3. 閾値を越えていない葉ノードについたら終局までランダムにシミュレーションを行う
4. プレイアウトの結果を辿ってきたノードに反映する

この流れを図に示したものが、図 1 である。白と黒の丸がノード一つ一つを表しており、実線の矢印がそのプレイアウトで選んだ枝を示している。このように一度のプレイアウトでゲームを終局までプレイし、その結果を返す。それぞれのノードが勝率とそのノードを通してプレイアウトした回数を保持しており、返ってきた結果に応じて、その勝敗数を更新する。この根ノードからのプレイアウトを何度も繰り返し、最終的には根の子ノードについて、勝率などの基準に基づいて最善手を決定する。

### 2.2 Upper Confidence Bound applied to Trees

UCT は、モンテカルロ木探索の一手法であり、Upper Confidence Bound (UCB) 値と呼ばれる値を用いて、子

ノードの選択を決定するアルゴリズムである。UCB 値は単純に有望そうな手に対して高くなるだけでなく、まだ充分調べておらず、有望である可能性が残されている手に対しても高くなる。そのため、満遍なく候補手を調べながら最善手を見つけることが出来る。

UCB 値は、以下の式で示される。

$$UCB = X_j + c \sqrt{\frac{\log n}{n_j}} \quad (1)$$

ただし、ここで用いられているパラメータは以下の通りである。

$X_j$  : j 番目の選択枝の報酬の期待値

$n$  : プレイアウト回数の合計

$n_j$  : j 番目の選択枝を選びプレイアウトした回数の合計

$c$  : 定数

式を見て分かる通り、報酬の期待値で表される選択枝の有望さと、プレイアウト回数によって表される選択枝の信頼性を足した値となっている。

$c$  はアルゴリズムの性質を決める定数で、これが小さすぎると有望だと思った手ばかりをプレイアウトし、他の場所に最適解があった場合見逃しやすくなる。大きすぎると満遍なくプレイアウトを行うため、木の展開が浅くなる可能性がある。

### 2.3 モンテカルロ木探索の複雑なゲームへの適用

既存研究として、モンテカルロ木探索をターン制ストラテジーやリアルタイムストラテジーに応用した例も存在する。

ターン制ストラテジーの研究として盛んなのは、Civilization を模した Freeciv というオープンソースゲームの研究である。このゲームは、都市を開拓し、技術開発を行い、国を発展させて戦争や外交を行う、複雑な要素が入ったゲームである。Freeciv の長期的戦略について、マニュアルを用いたモンテカルロ木探索を行う方法が提案されている [6], [7]。しかし、これらの研究では駒の複雑さを小さくするため、個々のユニットが独立して順番に動くことを仮定しており、戦術面では不十分と言える。

また、多人数の不完全ゲームであるカタンにモンテカルロ木探索を適用した例も存在する [8]。この研究では、カタンに関する領域知識を導入し、探索を制限することにより、効率のよい探索を行っている。

リアルタイムストラテジーゲームの分野でも、モンテカルロ木探索は用いられている [9]。本来、モンテカルロ木探索は、行動に同時性がないボードゲームなどに有効なものだが、リアルタイムストラテジーゲームはリアルタイムで互いのプレイヤーが駒を動かすゲームであり、行動は一定の時間持続し、同時性も存在する。これに対するアプローチとして、暇な部隊に対してのみ、他の部隊に合流するか、

敵を攻撃するかという選択枝を決定するためにモンテカルロ木探索を行う手法が提案されている。

### 2.4 ターン制ストラテジーに対するその他のアプローチ

ターン制ストラテジーに対するアプローチとしては、Freeciv を題材にした研究が盛んに行われている。これについて、例えば、Analogical Learning を用いて、Civilization における都市の資源活用を最適化する研究 [10] や、Q-learning を用いて、都市を開発する場所を最適化する研究 [11] がなされている。

前者は、類推を用いた学習方法で、個々の決定とそれに関する知識ベースのクエリ、そしてその評価を行うクエリを用意しておく。そして実行した決定に関して似通った事例を当てはめて評価を行う。その決定の事例のスナップショットを作成し、更に検索対象として追加することによって、学習を進める。しかし、本研究の場合、戦況が似通っているということの判断が難しく、この方法は向かない。

後者は強化学習の一種で、モンテカルロ木探索と似ているが、こちらはプレイアウトを行わず、報酬がわかるところから前の状態の値を更新させて伝播させて全体の状態の値を予測する。しかし、本研究では状態数が多く、全ての状態について情報を保持しているのは現実的ではなく、Q-learning は有効ではないと考えられるため、状態を保持せずに報酬を反映できるモンテカルロ木探索を用いた。

## 3. 評価用ターン制ストラテジーゲームの提案

本研究では、中国全土を一枚マップで表現し、国同士の戦争をモチーフにした、三国志 11 [12] を参考に、外交や内政といった要素を取り除き、戦闘のみに着目したゲームセットを作成する。ターン制ストラテジーゲームの研究においては、関連研究で紹介したとおり Freeciv がよく用いられるが、このゲームは戦闘よりも外交や研究、内政に重点が置かれたゲームであり、本研究のように駒一つ一つに着目する場合には適さないと考えたため、自作の環境での評価を行う。評価用に作成したゲームのルールを以下に示す。

- 駒は三つのパラメータ (攻撃力・防御力・知力) と、一つの特長能力を持つ。これはゲームの開始と共にランダムに設定される。三つのパラメータの上限はすべて 100 であり、下限は 1 である。また、それとは別に全ての駒は開始時に一律の HP を持っている。
- 完全情報ゲームであり、相手のパラメータ、HP、特長能力、座標など全ての情報はお互いに関示されている。
- プレイヤー一人につき 3 個の駒が与えられる。基本の移動力は 4 マスである。
- 駒は隣接する敵の駒に攻撃することが出来る。その時の HP へのダメージは自分の攻撃力と相手の防御力

によって変動する．攻撃した側も三分の一程度のダメージを受ける．HP がゼロになると駒は戦場から消滅する．

- ・ プレイヤーは 1 ターンに自分の持つ全ての駒を動かすことができる．行動終了を選ぶか、動かせる駒がいなくなると、相手に行動権が移る．
- ・ 敵と味方の行動が終わった段階で、1 ターン経過とする．
- ・ 勝ち負けは、以下のように決定する．
  - 「勝ち」は相手を全滅させた上で自分の駒が一つ以上無事である状態を表す．
  - 「負け」は相手が生き残っている上で全滅した状態を表す．
  - 「引き分け」は相手と自身がどちらも共倒れになった場合を表す．
- ・ マップの大きさは  $6 \times 7$  とした．
- ・ 敵の駒の上下左右 1 マス以内に進むと、駒の残りの移動力が強制的にゼロになる．
- ・ 特殊能力は、疾走、火計、剛胆、突撃、混乱の五種類からランダムに一つ与えられる．
  - 疾走は移動時に 1 マス多く動くことができる．
  - 火計は 2 マス離れたところからダメージを与えることができる特殊技が使える．このダメージは相手と自分の知力に依存し、反撃ダメージを受けない．
  - 剛胆は攻撃のダメージが 1.02 倍になる．
  - 突撃は、相手を 1 マス押し込む特殊技が使える．
  - 混乱は、2 マス離れたところから、相手を一定確率で 1 ターン行動不能にできる特殊技が使える．成功確率は相手の知力と実行する駒の知力によって左右される．
- ・ 一つの駒につき、移動は 1 ターンに 1 度しか行えない．また、攻撃と特殊技は、1 ターンに 1 回、どちらか片方しか行えない．
- ・ 攻撃を味方に当てることはできない．
- ・ 初期配置は図 2 に示す．

以上のようなゲームシステムを構築し、この中で評価を行う．

盤面の大きさと、移動範囲のサンプルを図 3 に示す．白い丸が自分の駒、黒い丸が敵の駒である．灰色に塗られているマスが、味方の駒が、疾走の特殊能力を持っておらず、移動力が 4 であるとした場合の、味方の駒の移動範囲となる．敵の上下左右（図において、x の記号がついているマス目）に移動してしまうと行動力が強制的にすべて消費されるため、この状況から敵の裏側に回り込むことはできない．

#### 4. 探索手法

本研究では、ターン制ストラテジーゲームに対する UCT

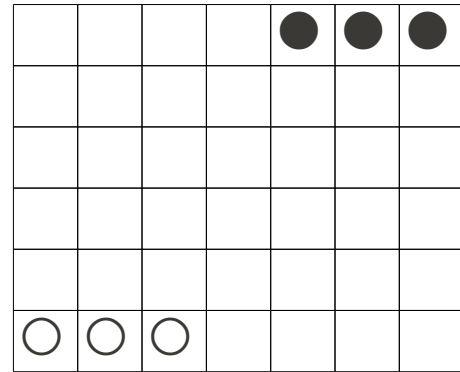


図 2 初期配置

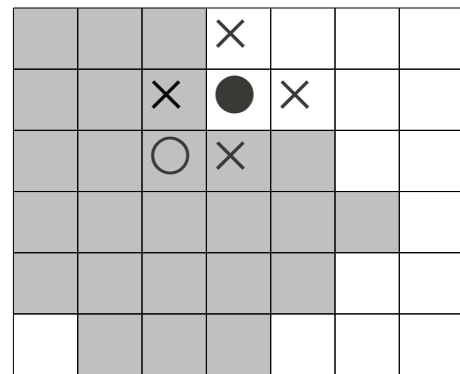


図 3 盤面と移動範囲

の適用と、それに伴う候補手の枝刈りを提案する．以降、モンテカルロ木の生成方法、プレイアウトの実施方法、候補手の枝刈りの方法についてそれぞれ説明する．

##### 4.1 モンテカルロ木の生成

ターン制ストラテジーゲームは将棋や碁と違い 1 ターンに多数の駒を動かすため、1 ターンが一つの行動とイコールではないため、モンテカルロ木探索のノードを作成するにあたり、二つの方法を提案する．まず一つは、一つ一つの行動を枝、すなわち選択枝として代入し、1 ターンを数段のノードに分けた縦長の木を作成する方法である．もう一つは、1 ターン内に可能な行動の組み合わせをすべて考え、その全ての組み合わせを作り、一つ一つを 1 ターン内の行動とする．そして、それに基づいて枝を作成し、1 ターンを一段のノードで表す方法である．

本研究では、前者を採用してモンテカルロ木を生成した．すなわち、枝は一つの駒が一つ行動を行うことを表しており、ノードは一つの行動ごとに変化した局面を表している．これにより、例え木の探索が不十分であっても、根ノードから出る枝の数が少ないため、取るべき次の一手に関して

は多くの情報を得られることができると考えられる。また、全ての手を一度に決めてしまう方法に比べて、弱い手を打ってしまった場合でも挽回可能である。

#### 4.2 プレイアウト

モンテカルロ木探索を実施する際に、プレイアウトはゲームの最後まで行われるべきものであることは前述した。しかし、ターン制ストラテジーにおいては、単純にモンテカルロ木探索を行うとあまりにも選択枝が多く、また、ターン数を重ねることで自動的に選択枝が狭まって行く囲碁やオセロと違い、永遠に勝負を続けることができってしまうため、確実にプレイアウトが終了する保証がない。そこで、無限ループに陥ることを防ぐために、プレイアウトをターン数に応じて強制的に終わらせる処置を取る。ターン数が過ぎてしまったときは、報酬として、勝ち負けではなく倒した駒の割合を返すようにする。例えば、敵の駒を3つのうち1つ倒した場合、 $1/3$ が報酬として返される。

#### 4.3 合法手の枝刈り

ターン制ストラテジーゲームの選択枝は、無駄なものが多いため、これらを枝刈りによって削減する。具体的には、ターン制ストラテジーの探索を膨大にしている原因の一つである移動について、制限を設ける。この制限のルールは以下の通りである。

- ・ 移動範囲の中で、敵の前後左右のマス
- ・ 移動範囲の中で、駒との距離が移動できる距離と同じであるマス

このようにすることで、敵から逃げる、迂回するといった大きな移動と、敵の周囲に接近する移動の二つを抽出することが出来る。本来はこれ以外の場所にいけることで可能になる戦略も存在するはずだが、枝刈りしなかった場所は多くの場合無駄になり、また、もしそのような戦略があっても発見することは難しいと考えられる。そこで、そのような戦略を無視してでも限られた有望そうな戦略について先の局面をより深く読むことで、限られた手の中でもっとも有望なものを信頼性高く見つけることができれば、信頼性の低い結果しか得られない通常のUCTに比べて良い手を見つけられると考えられる。その他、敵への攻撃も多数選択枝がある場合が存在するが、こちらは移動と比べて選択枝が少ないため、影響は少ないと考えて枝刈りは行わない。

また、本来は火計と混乱を持つ駒は敵の上下左右のマスだけでなく、周囲2マスを考慮に加える必要があると考えられるが、それでは結局行動の種類が多くなってしまうこと、また、距離の利点こそなくなるものの、1マス離れたところからでも特殊技は実行できることを考慮し、削除する。

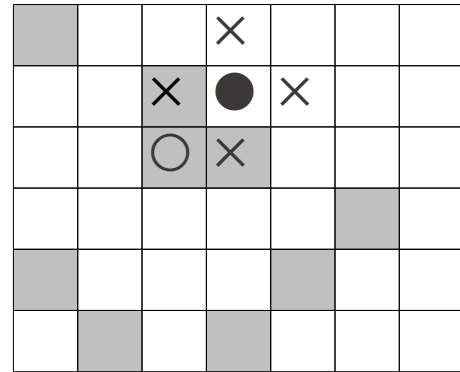


図 4 枝刈り後の移動範囲

## 5. 評価

### 5.1 評価設定

評価としては、提案手法に沿って選択枝を枝刈りしたUCTのAIを作成し、ベースラインとして作成したAIと対戦し、その比較で提案手法の評価を行う。手法の評価に用いたゲームの内容は、前述した通りである。このマップの移動範囲について、枝刈りを行った例を図4に示す。これは先ほど例示した図3の行動範囲を枝刈りしたものである。

ベースラインとして作成したAIは二つである。一つは、枝刈りを行わず、単純にUCTによって最適な候補手を探索するものである。もう一つは移動をランダムに行い、攻撃するだけのAIとなっている。ただし、移動前に敵に隣接している状況では、必ず攻撃を行ってからランダムに移動場所を決定する。移動先が敵に隣接しており、攻撃可能であればその場合も攻撃を行う。

また、プレイアウトの傾向を決める(1)式のcの値はどちらも同じく、0.15とした。これは経験的によい値を模索し、決定したものである。

今回は、一度の対戦にかかる時間を短縮するために、10ターン以内に決着がつかない場合、以下の基準に従って勝敗を決めた。

- 味方の駒の合計HPが相手に比べて勝っている場合、「勝ち」とする。
- 味方の駒の合計HPが相手に比べて少ない場合、「負け」とする。
- お互いの駒の合計HPが同じであった場合、「引分」とする。

### 5.2 対戦による評価

枝刈りUCTを単純なUCT、ランダムな移動+攻撃のAIと戦わせ、その勝敗を観察した。この時、UCTのプレイアウトを行う回数について、二つのルールを作り、勝率

が変わるかどうかについても検証した．そのルールを以下に示す．

- (1) 全てのパラメータを 50 にし，五つの中からランダムに特殊能力を与えた 3 つの駒を用い，対戦する．この時，プレイアウトの回数は 5,000 で固定とする．
- (2) 全てのパラメータを 50 にし，五つの中からランダムに特殊能力を与えた 3 つの駒を用い，対戦する．この時，プレイアウトは探索時間が 3 秒を過ぎるまで何回でも続けるものとする．

どちらのルールでも，対 UCT は 200 戦，対ランダム + ルールは 40 戦行った．

表 1，表 2 は，この二つの AI に対して，枝刈り UCT を条件を変えて戦わせた結果である．

表に示したように，いずれの結果についても選択肢を狭め枝刈りを行った UCT の方が勝ち越すという結果になった．簡易なルールベースの AI に対しては先攻，後攻問わず枝刈り UCT を用いた AI が全勝をおさめた．ランダム移動と攻撃を行うだけの AI では勝てない，戦略性のあるゲームとして，本稿で提案したゲームが作られていることが確認できたと言える．また，単純に UCT を行う AI に対しても，全勝はできなかったものの多くの勝ち星を挙げた．

### 5.3 詳細な動作の解析

枝刈りによって取れる戦術が狭まっている中，深く読むことによってどの程度戦術の不利を補っているのかを見るため，別の条件での動作の解析を行った．前述の対戦による評価と違い，パラメータもランダムに決定し，1 度だけ対戦を行い，枝刈りした AI がこれをどのように動かすかを観察した．ランダムに決定された駒のパラメータは表 3 の通りである．プレイアウトのルールは，(1) を使った．

その初手からの動きを表したのが，図 5，図 6，図 7 である．この図では，初手から駒 12 に対して駒 02 が火計を仕掛け，更にパラメータの一番低い駒 02 を囷として，次のターンに相手の最強の駒である駒 12 に対して駒 00 が攻

表 1 (1) のルールによる枝刈り UCT の勝率

対戦相手	枝刈り UCT の手番	勝ち	負け	引き分け
UCT	先攻	118	40	42
	後攻	136	34	30
ルールベース	先攻	40	0	0
	後攻	40	0	0

表 2 (2) のルールによる枝刈り UCT の勝率

対戦相手	枝刈り UCT の手番	勝ち	負け	引き分け
UCT	先攻	152	30	18
	後攻	154	22	24
ルールベース	先攻	40	0	0
	後攻	40	0	0

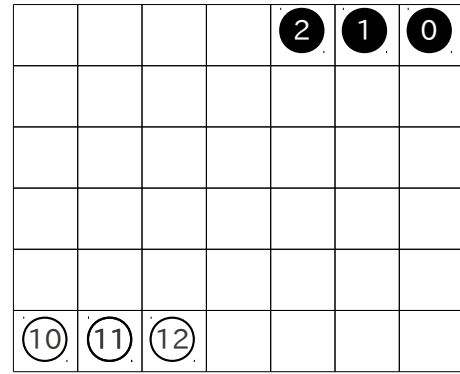


図 5 戦闘経過：先攻 1 ターン目

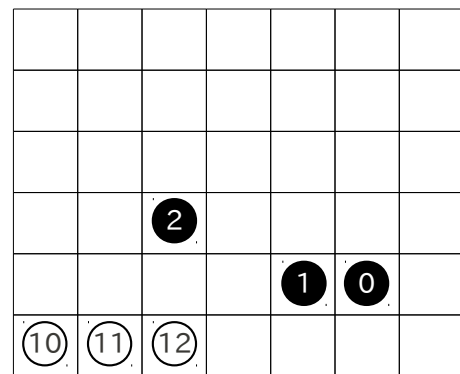


図 6 戦闘経過：後攻 1 ターン目

撃，駒 01 が火計を行い，HP を一気に削っている．このように，枝刈りによって行動範囲が絞られている中でも最適解に近い戦術をきっちりと発見し，行動に移していることが確認できる．

## 6. 考察

本項では，評価に関する考察と，改善するために行うべき，合法手の抽象化について説明する．更に，本研究で行っていた盤面の拡張に関して考察を加える．

### 6.1 評価に関する考察

まず，対戦による評価では，枝刈りを行った UCT が良い成績を収めた．枝刈りした UCT は合法手を減らしている

表 3 ランダムに決定されたパラメータ

プレイヤー	駒の番号	攻撃	防御	知力	特殊能力
枝刈り UCT	00	7	21	48	混乱
	01	4	97	71	火計
	02	5	51	17	火計
UCT	10	17	29	60	混乱
	11	10	28	61	混乱
	12	58	97	23	突撃

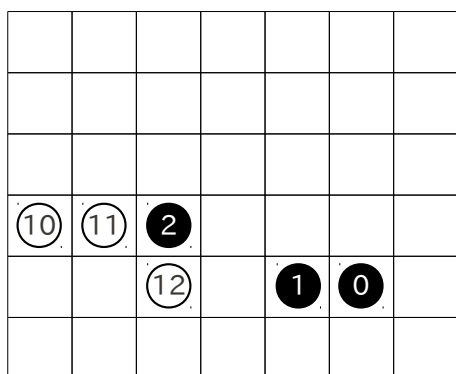


図 7 戦闘経過：先攻 2 ターン目

ため、それによって可能な戦術が少なくなってしまうデメリットも存在する。しかし、それ以上にモンテカルロ木を深く探索できる恩恵が大きく、このような結果につながったと考えられる。二つ目の実験では枝刈りをした移動先でも良い場所を見つけることができている、この考察を裏付けている。

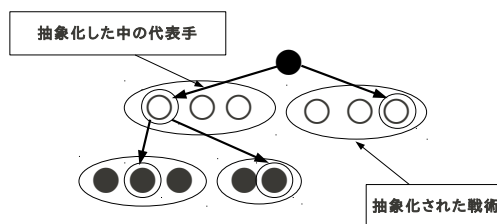
ただし、表 1、表 2 を見て分かる通り、引き分けが多く発生している。これは、主に攻撃した際の自分へのダメージによって、最後の二つの駒が共倒れになったことよって起きている。自分への反動の大きさについては、もう少しパラメータの調節が必要であると考えられる。

また、時間でプレイアウトを区切った対決の方が、枝刈りを行った UCT の勝率が高い結果となった。これは、枝刈りを行ったことによりプレイアウト中も無駄な選択枝を省いたシミュレーションを行ったことで、一度のプレイアウトに掛かる時間が減少し、一定時間内により多くのプレイアウトを行うことができたためであると考えられる。

## 6.2 合法手の抽象化

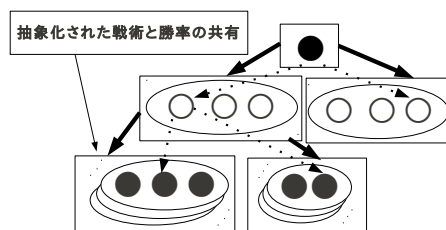
現状の枝刈りの問題点として、移動場所の限定しか行っていないため、攻撃することを前提として敵の前後左右に移動するにも関わらず、攻撃しないというような、次の行動との兼ね合いで無駄になる選択枝を削除できていないことが挙げられる。また、大きく動く場所のみを移動の候補に入れたが、実際には敵から離れる方向であればどこに動いても展開が変わらないことも多く、これもまだ改善の余地があると言える。そこで、これらの問題点を改善するために、ただ単純に枝刈りを行うのではなく、選択枝を抽象化して纏めるという方法がある。これに関して、二つの方法が考えられる。

一つは、合法手を抽象的な選択枝に分類し、その中から代表的な一つを選び出して UCT のノードとする方法である。一見すると枝刈りと変わらないが、無駄な選択枝を削っているのではなく、同様の意味を持つ選択枝を一つに



(1) 抽象化した中で代表手を選び、ノードに入れる手法

図 8 代表の選択枝を選ぶモンテカルロ木



(2) 抽象化した選択枝群を同じノードに入れ、プレイアウト時はランダムに選ぶ手法

図 9 抽象化された戦術で勝率を共有するモンテカルロ木

纏めているのであり、抽象化したと言える。ただしこの方法では、代表となる選択枝の選び方に大きく左右される AI となってしまう可能性がある。

もう一つは、合法手を分類した上で、その勝率をすべて UCT の一つのノードとして管理する方法である。UCT を探索する際には、同じノードになっている選択枝からランダムに選択し、プレイアウトを行う。こうすることで、どの方針が上手く行くかという指針を得ることが出来る。実際の手は、選ばれた分類の中から代表的なものを選べば良い。

この二つの方法によって作られるモンテカルロ木を示したものが、図 8 と図 9 である。

## 6.3 広大な盤面における盤面の縮小

また、本研究ではごく小さな盤面において戦闘を行っている。しかし、一般的なターン制ストラテジーゲームの盤面は本研究で用いた盤面に比べ広いことがほとんどである。そのため、探索空間は更に膨大になる。これを改善し、本研究と同様の条件で扱えるようにするために、大きな盤面から戦闘を行っている局所的な戦場を抽出することが必要

となる。この時、戦場から外れる部分についての探索は不十分になるが、ターン制ストラテジーゲームの駒は盤面全体に影響を与えるような手をわずかししか持たないことが多い。そのため、戦場を分けることによるデメリットは僅かであると考えられる。この盤面の縮小について、二つの方法が考えられる。

一つは、最初から敵と味方が遭遇した付近の盤面を戦場として抽出し、固定してしまう方法である。大きめにとっておけば戦闘中に敵味方がそこを出ることは少ない。いちいち戦場の大きさを更新しなくてよいため、処理が高速に出来る。もう一つは、敵が次ターンで移動し、攻撃が可能である範囲内を、戦場として抽出する方法である。確実に最小の戦場を抽出することができる一方で、ターンごとに戦場の大きさを更新する必要があるため、処理が低速になる。

その他の問題点として、勝利が完全に確定してしまったとき（すなわち、相手から攻撃されてもカウンターで相手が倒れるなど、どの手を選んででも勝ちが決まっているとき）に、どのプレイアウトでも勝利してしまうために探索が全く意味をなさず、結果、攻撃せずにいたずらにターンを消費するといった現象も起こった。これを解決するため、早いターン数で倒した場合には報酬にボーナスをつけるなど、プレイアウトの報酬の修正をする必要があると考えられる。

## 7. まとめ

本研究では、選択枝が多数存在するターン制ストラテジーゲームの戦闘における探索空間を縮小する方法として、ターン制ストラテジーゲームの戦闘要素に対するUCTの適用方法と、移動時の選択枝を枝刈りすることを提案した。また、その評価方法として戦闘のみを抽出したゲームを作成した。これを評価するための基準として、単純にUCTを実行するだけのAIと、ランダムに移動し攻撃可能ならば攻撃を行うだけのAIを作成し、これらとの対戦を行った。その結果、ランダムなAIに対しては圧勝し、また、単純なUCTに対しても好成績を収めた。枝刈りによる探索空間の縮小が、選択枝を減らしてしまうデメリットを補い、上回ったと考えられる。また、詳細な動作の解析においても、枝刈りを行ってなお十分に戦略的な動きをしていることが確認できる結果となった。

本稿では、手の枝刈りが、ターン制シミュレーションのような局所性のある問題に対して有効であることを示した。しかし、これはまだ無駄な手を省いただけにすぎず、これ以上に手を抽象化して更に深い探索を行うことが可能なのである。そこで、今後の課題としては、合法手の抽象化と盤面の縮小を行い、有効性を検証していきたい。更に、最終的には合法手の抽象化を自動で行っていくAIを目指したい。

## 参考文献

- [1] 「タクティクスオウガ 運命の輪」<http://www.square-enix.co.jp/tacticsogre/>.
- [2] 「シヴィライゼーション 4【完全日本語版】- サイバーフロント」<http://www.cyberfront.co.jp/title/civ4/>.
- [3] 「Battle for Wesnoth」<http://www.wesnoth.org/>.
- [4] Kocsis, L., Szepesvári, C.: Bandit based monte-carlo planning, *In Machine Learning: ECML 2006*, Springer, pp. 282–293 (2006).
- [5] 美添一樹, 山下宏. コンピュータ囲碁-モンテカルロ法の理論と実践-. 共立出版, (2011).
- [6] Branavan, S., Silver, D. and Barzilay, R.: Non-linear monte-carlo search in civilization ii, *Proceedings of the Twenty-Second international joint conference on Artificial Intelligence-Volume Volume Three*, AAAI Press, pp. 2404–2410 (2011).
- [7] Branavan, S., Silver, D. and Barzilay, R.: Learning to win by reading manuals in a monte-carlo framework, *Journal of Artificial Intelligence Research*, Vol. 43, No. 1, pp. 661–704 (2012).
- [8] Szita, I., Chaslot, G. and Spronck, P.: Monte-carlo tree search in settlers of Catan, *Advances in Computer Games*, Springer, pp. 21–32 (2010).
- [9] Balla, R.-K. and Fern, A.: UCT for Tactical Assault Planning in Real-Time Strategy Games., *Proceedings of the Twenty-First International Joint Conference on Artificial Intelligence*, pp. 40–45 (2009).
- [10] Hinrichs, T. R. and Forbus, K. D.: Analogical Learning in a Turn-Based Strategy Game., *Proceedings of the Twenty-First International Joint Conference on Artificial Intelligence*, pp. 853–858 (2007).
- [11] Wender, S. and Watson, I.: Using reinforcement learning for city site selection in the turn-based strategy game Civilization IV, *the 2008 IEEE Symposium On Computational Intelligence and Games*, IEEE, pp. 372–377 (2008).
- [12] 「三国志 11-GameCity」<http://www.gamecity.ne.jp/sangokushi/11/>.