難しさが手番で異なる局面での モンテカルロ木探索の性能の改善

今川　孝久[1]　金子　知適[1]

概要：モンテカルロ木探索 (MCTS) は広い応用を持つ探索手法であり, 特に囲碁で効果的であることが知られている. 一方, 置き碁や攻め合いなど苦手とする局面があるように，MCTS が有効に働く条件はまだはっきりと分かっていない．本研究では「最善手を互いに選び続ければ引き分けになるが, 次善手を選んだ場合の不利益の度合いが手番によって異なる」という状況を詳しく分析した．まず，この性質を持つゲームでは MCTS が最善手を逃しやすいことが，仮想ゲームを利用した分析を通じて分かった．そして改善する鍵が，終局時のスコアを利得に変換する調整にあることを示し，プレイアウト終了時のスコアの頻度が最大の位置に勝ち負けの評価の境界を定める手法を提案した．実際に対局実験を行ったところ，提案手法を MCTS に用いることで，通常の MCTS や別の調整法である dynamic komi を用いる場合よりも高い勝率が得られた．

Improvement of Performance of Monte Carlo Tree Search in Positions Where Difficulty Differs by Turns

Takahisa Imagawa[1]　Tomoyuki Kaneko[1]

Abstract: Monte-Carlo tree search (MCTS) is famous for its success in Go, while it is widely applied in many search problems. However, it is not clear that what kind of game conditions should be satisfied so that MCTS works effectively. For example, it is known that MCTS programs often play awkward moves in positions involving semeai and ones in handicapped games. This paper analyzes and discusses a property in even positions where two players have different penalties when they missed the best move. Analyses based on virtual games show that MCTS often fails to identify the best move in this situation and that the problem will be mitigated by adjusting a function that converts game scores gained by playouts into rewards (win, draw and loss). We presented to use a score of the maximum frequency as a threshold in the function. Experiments show that MCTS with the presented method works better than plain MCTS and MCTS with dynamic komi.

1. はじめに

モンテカルロ木探索 (MCTS) は，囲碁などの探索空間が大きなゲームでも，概ね効果的に動作する優れた探索手法である．MCTS が提案される前はチェス等で効果的だった MinMax 探索を基本とした探索手法が囲碁プログラムで用いられていた．しかし，囲碁では良い評価関数を作るのが難しく，MinMax 探索ではなかなか強くならなかった．MCTS では，ランダムに終局まで手を打つというプレイアウトを繰り返して，どの手が良いかを探索する．途中の局面で評価が難しいゲームでも，終局では，ルールからスコア (目数) を計算できる．主としてその性質により，MCTS は囲碁で効果的であると考えられる．

その一方で MCTS は，囲碁の局面の中でも，一定の性質を持つ局面では本来の性能が出にくいことが知られてきている．例として置き碁や攻め合いが挙げられる．

置き碁の場合の難しさはスコアの期待値が 0 から大きく離れていることが原因と予想される．このような状況では，MCTS プレイヤは最善手を打つことが難しく，不利な側のプレイヤは相手のミスを期待した無謀な手を，有利な側のプレイヤは最善でない無難で，緩んだ手を打ちやすくなる．dynamic komi[2] はこの問題に対して，有利・不利が釣り合う方向にスコアを補正する手法である．

[1] 東京大学大学院総合文化研究科
Graduate School of Arts and Sciences, The University of Tokyo

攻め合いの難しさは，スコアの期待値にあまり差が無い，通常の手と，スコアの期待値が大きく動く手の両方が存在する点にあると予想される．囲碁で石の集団を取るような手は，スコアの大きく動く手の例である．そのような場合，根の候補手の間の価値の小さな差は相対的に小さくなる為，評価出来なくなる恐れがある．別の原因には，悪い手のスコアの期待値が探索初期に高いということも考えられる．改善の前段階として，攻め合いを見つけ出すといった研究 [6] はなされているが，攻め合いについては今のところ，効果的な改善策は見つかっていない．

本研究では，MCTS が上手く働かないゲームの性質を新たに示す．それは，「互角でかつ最善手と次善手の価値の差が手番によって異なる局面」を持つゲームである．仮想ゲームとして定義し，分析を行った．仮想ゲームを用いる利点とは，ゲームの途中の状態に対する勝敗の理論値や確率的に着手する場合の勝敗の期待値を求めることができることである．それらを用いることで，正確な分析が可能となる．

最善手と次善手の価値の差とは，プレイヤが間違えた時の，すなわち最善手を逃した場合の被害の大きさに対応する．たとえば，先手はどの手を指しても大きな差は無い一方で，後手は間違えがすぐに負けにつながるという局面であれば，仮にゲームの理論値は引き分けであっても後手の勝ちにくい局面と言える．本研究で用いる局面はそのような特徴を持っているので，攻め合いの難しさの一因と予想される，「大きな価値の手が存在し，根の候補手の価値の小さな差の評価が難しい」という状況に近い．また，一方が有利な局面という置き碁とも共通する性質を持つ．そこで，dynamic komi との性能の比較も行う．置き碁では，下手のスコアの MinMax 値は正に偏っているはずであり，少し緩んだ手を打っても，その後最善手を打ち続ければ，上手の手に依らず，勝てる (自分が上手の場合，相手にミスがなければ勝ち目がない)．一方で本研究で扱う，MinMax 値が偏っていない場合では，一度でも最善手を逃すと，その後最善手を打ち続けても，相手が最善手を打ち続ければ負けるという違いがある．

このような状況での MCTS の性能を改善するために，プレイアウト結果の頻度最大となるスコアを仮想的に勝ち負けの境界とした手法を提案した．実験の結果この手法は，従来の MCTS が上手く行かない仮想ゲームでも効果的であった．

2. 背景

本研究では有利不利が偏った仮想ゲームでの, MCTS の性能を改善することを目指している．この節では MCTS の有力な一手法である UCT や，偏った状況下での UCT の改善策である dynamic komi，また仮想ゲームの性質と UCT の性能について調べた研究を紹介する．

2.1 UCT

UCB1[1] は多腕バンディット問題 [5] に対するアルゴリズムの一つである．この問題では，各選択肢に対して，確率分布が決められていて，ある選択肢を選ぶとその利得が確率的に得られる．そのような状況下で，決められた回数，選択肢を選んだ場合の利得の総和を最大化するという問題である．選択肢 i の利得の期待値を μ_i とする．その内，利得の期待値が最大のものを $*$ と表し，その期待値を μ_* とする．毎回の選択で，最適な選択肢 $*$ を選び続ければ，目的は果たされるが利得の分布を知らずに選ぶ必要があるため不可能である．例えば，今現在で最も利得の平均が高い選択肢を選ぶ方針は有力ではあるが，その選択肢が最適な選択肢でない場合は，最適な選択肢を選び続けた場合と比べて平均的には損となる．一方で各選択肢を丹念に調べれば，どれが最適な期待値を持つか，より情報が増えるが，利得の最大化という目的から，見込みの薄い選択肢を多数試行することは望ましくない．UCB1 アルゴリズムでは，有望さとその評価の確かさを組み合わせた評価基準である UCB 値を定義し，それが最大となる選択肢を選ぶ．つまり，取りうる選択肢の集合を A, 選択肢 j を通った場合の平均利得を $\bar{r_j}$ その選択回数を n_j とすると

$$\arg\max_{j \in A} = \mathrm{UCB}_j \equiv \bar{r_j} + \sqrt{\frac{2\log(s)}{n_j}}. \qquad (1)$$

という j を毎回選択する．平均利得 $\bar{r_j}$ により，有望な選択肢が，それに加えられる補正項により，評価が不確かなものが選ばれる．

UCB1 の効率を議論するために，$\Delta_i \equiv \mu_* - \mu_i$ という選択肢 i を選んだ際に，最適な選択肢と比べて平均としてどれくらい損をしたか，という値を考える．UCB1 アルゴリズムでは n 回試行中，選択肢 i を選ぶ回数の期待値 $\mathbb{E}[T_i(n)]$ に対して，

$$\mathbb{E}[T_i(n)] \leq \frac{8\ln(n)}{\Delta_i^2} + 1 + \frac{\pi^2}{3} \qquad (2)$$

という不等式が成り立つ．このような知見を MCTS に応用した手法が UCT である．UCT[4] では探索木のどの葉からプレイアウトするかの決定を多腕バンディット問題の変種とみなして，UCB1 をその決定に用いる．UCT ではまず，根から UCB 値が最大の手を繰り返し葉に到達するまで選んでいく．葉に到達したら，プレイアウトを行う．尚その節点を訪問した回数が閾値を超えていたら探索木を成長させる．そして，プレイアウト結果 (利得) を親・先祖に伝えていく．それを繰り返し行うことで評価の精度を高めていく．UCT でも，式 (2) に相当する式があり，様々な仮定を置いた上，$\mathbb{E}[T_i(n)]$ に対して Δ_i の自乗に反比例する上限が知られている．そして，探索終了後にどの手を最善手として選ぶかの方法には何種類かあるが，「最も訪問した回数の多い節点を最善手として選ぶ」という方法が一般的で，本研究でも採用した．また，囲碁ではスコアそのものよりも，勝敗に

Algorithm 1 SCORE SITUATIONAL

if MoveNum $<$ 20 **then**
  Komi $\leftarrow$ LinearHandicap
  Rachet $\leftarrow \infty$
  return
end if
BoardOccupiedRatio $\leftarrow \frac{\text{OccupiedIntersections}}{\text{Intersections}}$
GamePhase $\leftarrow$ BoardOccupiedRatio $+s$
KomiRate $\leftarrow (1 + \exp(c \cdot \text{GamePhase}))^{-1}$
Komi $\leftarrow$ Komi + KomiRate $\cdot \mathbb{E}$[Score]

Algorithm 2 VALUE SITUATIONAL

if MoveNum $<$ 20 **then**
  Komi $\leftarrow$ LinearHandicap
  Rachet $\leftarrow \infty$
  return
end if
if Value $<$ red **then**
  if Komi $>$ 0 **then**
    Rachet $\leftarrow$ Komi
  end if
  Komi $\leftarrow$ Komi - 1
else
  if Value $>$ green $\wedge$ Komi $<$ Rachet **then**
    Komi $\leftarrow$ Komi + 1
  end if
end if

変換して UCT で扱う方が強いことが知られている．そこで本研究でも全てのアルゴリズムでプレイアウト結果を勝ち：1，引き分け：0.5，負け：0 に利得を変換して用いた．

2.2 dynamic komi

置き碁のように一方のプレイヤが有利な状況では，UCT が本来の性能を発揮できないことが知られている．良い手を指し続ければ，有利な場合勝ちをより確実なものにでき，不利な場合負け確実から拮抗した状態に持っていくことができる可能性があっても，良い手を指せない．そのため，有利な状態から互角に，不利な状態からより不利な状態になってしまうことがある．

この理由は明らかである．有利な場合はどんな手を打ってもプレイアウト結果はほぼ勝ちで，負けることはどんな手を打っても有り得ないと判断することによる．そのため，大損は避けても少し損する手を選びやすくなる．不利な場合はどんな手を打ってもプレイアウト結果はほぼ負けで，勝つとすれば相手が重大なミスをする場合しか有り得ないと判断してしまいやすい．そのため，相手が有り得ないミスをすれば大得できるが，大抵そうでないので損をする，そのような手を選びやすくなる．一般に，一方のプレイヤが有利だと，プレイアウト結果が勝ちばかり，負けばかりとなって，プレイアウトで得られる情報量が少ないと言える．

dynamic komi はそのような問題への対処として提案された手法である．dynamic komi では，仮想的にコミを調整することで勝ち負けの境界をずらし，勝ち負けの頻度が同じぐらいになるようにして，プレイアウトで得られる情報量を増やす．ずらす方法には大きく分けて 2 種類が提案されている (参考文献 [2] 参照)．一つは，置き石の数に対して予め決められたコミを手数に応じて線形に加える方法と，もう一つは，それに加えて，プレイアウト結果に応じたコミ加える方法である．本稿の実験では置き石に相当するものは無いので前者は使わなかった．また，後者は手数の少ない時は前者を使うが，それも省いた．後者の方法にはさらに，スコアに応じて変える Score Situational と，利得に応じて変える Value Situational の 2 通りある．尚，dynamic komi を使うプレイヤは黒番と仮定している．

Score Situational では GamePhase というゲームの進行状態をシグモイド関数で変換して KomiRate という補正速度を定める．そして，スコアの期待値 (補正後) に KomiRate を掛けたものを Komi に足して，新たな Komi とする．このようにして補正後にスコアの期待値を 0 に近づけることを目指す．Value Situational は，利得の平均を区間に分けて，利得平均が green を超えたら (勝ちが多いので)Komi を増やし，逆に red を下回ったら (負けが多いので)Komi を減らすことが主な動作である．それに加えて，Rachet という Komi の上限を設け，Komi のむやみな上げ下げを抑えている．

論文 [2] 中では，Score Situational, Value Situational のパラメータとしてそれぞれ $s = 0.75, c = 20$, red $= 0.45$, green $= 0.50$ が用いられている．尚，前者では，s, c を十分大きくとり，後者では，red を 0 に green を 1 に取ることで通常の UCT と同じ振る舞いになる．

2.3 仮想ゲーム

ゲームの性質が MCTS の性能にどのように影響するかを調べた研究として，文献 [3] がある．この研究では囲碁等のゲームと同様，二人零和有限確定完全情報ゲームに属する仮想ゲームを提案し，パラメータの調節により探索空間の大きさを変更した際の，MCTS の性能への影響を示している．まず，仮想ゲームのルールを説明する．ゲームでは**図 1** のような盤を用いる．ゲームの初期局面では駒は S (Start) に置かれる．各手番では駒を 1 つ前に進め，その際，横に並ぶどの升を選んでも良い．各プレイヤは順番に手を指し，共に G (Goal) に達したらゲームが終了する．各升には秘密のペナルティが割り当てられていて，指した手 (選んだ升) に対応したペナルティが科せられる．ゲーム終了時，各自の踏んだペナルティの和が少ない方を勝ちとする．尚，各プレイヤのペナルティはゲーム中は隠されており，ゲーム終了時に初めてスコアが明かされる．仮想ゲームでのスコアは初期局面から各プレイヤが踏んできた升のペナルティの総和の差である．つまり，スコア = (相手のペナルティの総和) - (UCT プレイヤのペナルティの総和) となる．

次に例を挙げる．本稿で使う仮想ゲーム (図 1) では，各局面に最善手が一つだけあり，それ以外の手を選んだ

G	x	x	x	x	x	x	x	S
0	1	1	1	x	1	1	1	0
0	1	1	1	x	1	1	1	0
0	1	1	1	x	1	1	1	0
S	x	x	x	x	x	x	x	G

図 1 対称な仮想ゲームの例

G	x	x	x	x	x	x	x	S
0	1	1	1	x	4	4	4	0
0	1	1	1	x	4	4	4	0
0	1	1	1	x	4	4	4	0
S	x	x	x	x	x	x	x	G

図 2 非対称ゲーム (一律) の例

場合にどのような不利益があるのかをモデル化したゲームである．“x” は壁を意味し，そこには移動・通過できないことを意味する．このゲームでは，ペナルティが 0 の手が常に存在し，最善の戦略はその手を打ち続けて引き分けとすることである．それ以外の手のペナルティは 1 である．

このようなゲームを分析に用いることで，探索空間の大きさを簡単に変えながら実験することができる．また，プレイヤに利用可能な情報である，プレイアウト終了時のスコアとそれに伴う勝敗以外に，分析者は途中の各局面でのペナルティを利用して分析を行うことが出来る．対局においては最善手を常に選ぶプレイヤ (最適プレイヤと呼ぶ) を対戦相手とし，相手のミスにより偶然勝つようなことがなく，正しい手を選べたかどうか測定できる．

Finnsson の研究では探索空間が同じでもペナルティの設定次第で難しさが異なることを実験的に示した．尚，プレイアウト数は，探索木内プレイアウト中合わせた節点の訪問数を一定 (5000 回) にすることで決めている．これは恐らく，実時間でプレイアウト数を決めるのと似た効果を得るための工夫である．一つのノードを訪問するのにかかる時間が一定と仮定すれば，一定の時間でプレイアウト数を決めるのと等しくなるという性質がある．本研究でもこれに倣い，対戦相手を最適プレイヤとし，節点の訪問数を 5000 回としている．以後この種のゲームを対称ゲームと呼ぶことにする．

便宜のため，盤の内，ペナルティが与えられている列の長さを盤の長さ，行の長さを盤の幅とそれぞれ呼ぶことにする．この盤 (図 1) を例にすると長さ 3 幅 4 ということになる．

3. 分析手法

3.1 非対称な仮想ゲームの提案

この小節では各手法の得手不得手を分析する対象として，手番によって難しさが異なる二種類の状況を提案し，具体的な仮想ゲームとして定義する．

3.1.1 非対称ゲーム (一律)

このゲームは後手のプレイヤのペナルティを大きく設定したものである．その大きさに，一回でもペナルティを持つ手を選ぶと負けが確定する値を用いる．具体的には後手番のペナルティは 0 と (盤の長さ) + 1 とした．一方，先手のペナルティは図 1 の対称ゲームと同じままとする．長さ 3 幅 4 の盤の例を**図 2** に示した．ペナルティの差から後手にとっては勝ちにくいゲームであるが，初期局面の MinMax 値は依然として 0 である．実験では UCT プレイヤを先手とした．このゲームで長さ 6 幅 4 の盤で初手を UCT 探索をした場合のプレイアウト結果 (スコア) のヒストグラムをとると**図 3** のようになる．頻度は安定させるため，1000 回の探索を通じて合計し，プレイアウトは計 416000 回行なっている．

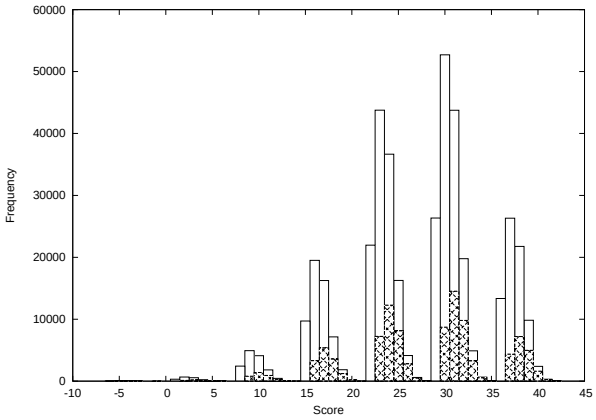

図 3 非対称ゲーム (一律) での根節点とその最善手のノードでのプレイアウト結果 (スコア) のヒストグラム (網掛けが最善手)

図 3 を見るとよく分かる通り，スコアが 0 以下になる頻度は，他に比べて非常に低く，最も頻度が高くても合計 38 回である．つまり，このゲームではプレイアウト結果がほとんど勝ちになってしまい，各手の評価に差がほとんど付かない．この点が UCT で最善手を選ぶ難しさと考えられる．このゲームでは UCT プレイヤの勝ち以外の結果が出るのは，相手が探索木中，プレイアウト中にペナルティ 0 の手を選び続けた場合だけであり，一様分布ではなかなか起こりえない．また，根直下の各節点を訪問したプレイアウト結果 (利得) に差が付くのは，相手が一度もミスをせず，なおかつ UCT プレイヤも根直下の手を除いて，常に最善手を選んだ場合に限られる．その場合だけは，根直下の最善手を選んだ場合は引き分けで利得 0.5，最善手以外の場合は負けで利得 0 となり，差が付く．

対称なゲームと比べた難しさを具体的に議論するため，対称・非対称ゲームで対戦実験を行った結果を示す．尚，本稿で行った実験では常に UCT プレイヤに対して常にペナルティが 0 の最善手を選ぶ最適プレイヤを対戦相手とした．また，UCT プレイヤは，先手でかつ有利な盤 (図 2 での左の盤) を使う．実験は 1000 試合行い，盤の幅 4 として，長さを変えた場合の勝率の変化を**図 4** のグラフに掲載する．非対称ゲームでは対称ゲームの場合

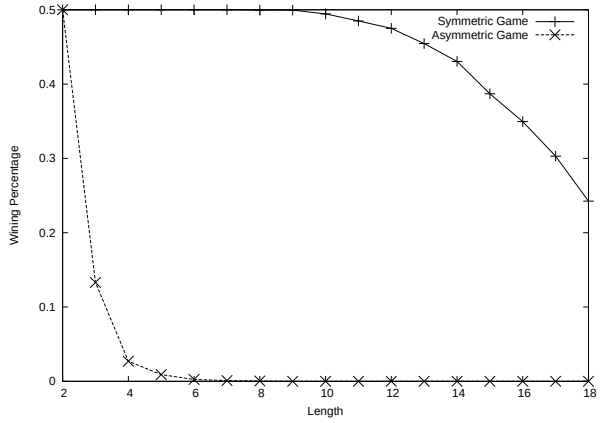

図 4 対称・非対称ゲームで盤を長くした場合の UCT アルゴリズムの勝率の変化

G	x	x	x	x	x	x	x	S
0	1	1	1	x	1	1	1	0
0	1	1	1	x	1	1	1	0
0	1	1	1	x	11	11	11	0
S	x	x	x	x	x	x	x	G

図 5 非対称ゲーム (最終手) の例

と比べて UCT の性能が明らかに低い．長さを 6 以上にすると，勝率がほとんど 0 となる．長さが短い場合に，勝率が高い理由としては主に 2 つ考えられる．1 つは読みきれることであり，もう 1 つは読み切ることが出来ない場合でも，勝ち以外のプレイアウト結果となる確率は盤が短いほうが高いことである．しかし，長くするにつれて，そのようなこともなくなり勝率が下がると予想される．

このゲームはプレイアウトの勝敗が偏っており，この状況の緩和策としては，勝敗の境界の補正が有効であると期待される．境界の補正に関する，dynamic komi という既存の手法との比較は実験の節で議論する．

3.1.2 非対称ゲーム (最終手)

別のゲームとして，後手のプレイヤの盤の一番最後の行に一つの升を除いて，非常に大きなペナルティを与え，それを踏むかどうかでスコアが大きく異なるようにしたものを考える．UCT で探索する場合には，最終手でのペナルティの有無によってプレイアウト結果のスコアは大きく変化し，ヒストグラムが二分されるという効果を持つ．そのためには，最終手のペナルティは 2·(盤の長さ) あれば十分である．しかし，頻度が 0 の部分をある程度作るため，2·(盤の長さ) + 5 とした．実際に盤の長さ 6 で幅 4 の時に，初手を UCT で探索して，スコアのヒストグラムを取ってみると**図 6** のようになり，確かに山が分かれている．このようなゲームでもスコア補正が有効な対策となりうる．なぜなら，補正無しの場合，最終手で踏むペナルティが 0 の場合に限り，プレイアウト結果が勝ちや負けに分かれる．一方，最終手で踏むペナルティが 0 でない場合，プレイアウト結果は勝ちにしかなり得ず，(幅 - 1) 手分の情報を捨てていると言えるからである．補正を行い，最終手で踏むペナルティが 0 でない場合で，プレイアウト結果は勝ちや負けに分かれるようにすれば 1 手分の情報を捨てるだけで済む．従って，補正をすることにより性能向上の余地がある．しかし，既存手法である dynamic komi ではサンプルの平均値付近に境界を移動する性質がある為，ヒストグラムで頻度が 0 の所 (図 6 のスコア 6 から 10) に境界が移動し，最善手とそうでない手を見分けることが難しいと予想される．

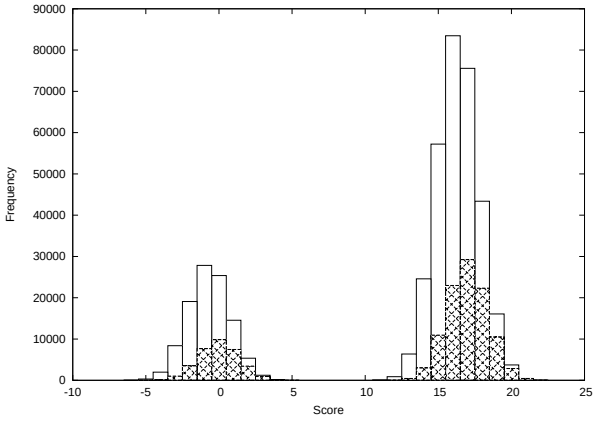

図 6 非対称ゲーム (最終手) での根節点とその最善手のノードでのプレイアウト結果 (スコア) のヒストグラム (網掛けが最善手)

3.2 新たな補正手法の提案

この小節では新たな補正方法を 2 種類提案する．一つは，プレイヤには入手できない情報を利用した方法である．これを比較のための目標値として用いるためのものである．もう一つは，プレイヤに入手可能な情報だけを利用した，現実的な利用を想定した手法である．

3.2.1 UCB1 での最適補正量

UCT でなく，1 手読みの UCB1 を用いた場合にどのような補正量が良いか考える．手の利得の期待値 μ に微小な差しかない場合と最善手の期待値 μ_* が他のものよりずっと大きい場合があったとする．それぞれで探索して最善手を選ぶ場合には，後者の方が簡単であると期待される．式 (2) のように，後者の方が最善手に計算資源を投入する強い保証を持つためである．この点を踏まえて，$\min_{i \in A \setminus \{*\}} \Delta_i \equiv \Delta$ を最大化する補正を提案する．Δ を最大化する補正量を求めるという問題は，プレイアウト開始節点毎のスコアの確率分布が分かれば解ける．そのため，本研究で扱うゲームではプレイヤに分からない知識 (手のペナルティの大きさは幾つか，またそのペナルティを持つ手は何個あるか) を使って求められる．

実際に非対称ゲーム (一律) で盤の長さ 6，幅 4，初手探索時に，補正量を変えた場合の Δ の値を 0.01 間隔でプロットした結果を**図 7** に示す．この例では多峰になること，Δ が 0 になる補正量があること，そして Δ は補正量 a に対して $30 < a < 31$ で最大になることが見て取れる．

このような関数の Δ 最大となる補正量 a を求めるため

には，ペナルティと盤の長さに比例する個数の非整数値を調べれば十分であることを以下で示す．具体的には，次善手のペナルティを p とし，最善手の節点からプレイアウトした場合のスコアの最小値を MinScore，最大値を MaxScore とすると，MaxScore $-\,p-$ MinScore $+\,2$ 個の非整数値について Δ の値を調べれば十分である．尚，MinScore $<$ MaxScore $-\,p$ かつ $p>0$ であること，また，ペナルティとスコアは整数値しか取らないことを仮定する．

本研究で扱っているゲームではプレイアウト中にペナルティを課される確率はそれ以前にどの升を踏んだかに依らないという性質がある．そのため，最善手の節点からプレイアウトを始めた場合のスコア s となる確率を $\mathrm{P}[s]$ とし，次善手のペナルティを p とすると，次善手の節点からプレイアウトした場合のスコア s となる確率は $\mathrm{P}[s+p]$ となる．

関数 $\mathrm{P}(x)$ を x が整数の時に $\mathrm{P}[x]$，非整数の時に 0 をとるものと定める．補正量 a での Δ の値を $\Delta(a)$ と表記すると $\Delta(a)=\mu_*-\mu=\left(\sum_{a<s}\mathrm{P}[s]+0.5\mathrm{P}(a)\right)-\left(\sum_{a<s}\mathrm{P}[s+p]+0.5\mathrm{P}(a+p)\right)$ と表せる．但し，この式では，利得がプレイアウト開始節点によって異なっていない (従って，打ち消し合う) 所も足している．式変形をすると，

$$\Delta(a)=\sum_{a<s<a+p}\mathrm{P}[s]+0.5\cdot\mathrm{P}(a)+0.5\cdot\mathrm{P}(a+p)\quad(3)$$

となる．

n は整数で，$n<a<n+1$ とする．任意の δ に対して $n<a+\delta<n+1$ ならば $\Delta(a)=\Delta(a+\delta)$ も成り立つ．これは，スコアが整数しかとりえないので，非整数値での微小な補正では，利得が変化することはないということによる．このことから，Δ の最大値を求めるにあたって，非整数補正での Δ の値は隣り合う整数の間で一つだけ調べれば十分であることが分かる．また，$\Delta(n+\delta)$ と $\Delta(n-\delta)$ が最大値である時のみ $\Delta(n)$ が最大値であること (n は整数，$0<\delta<1$) も成り立つ．このことから，補正量は非整数値のみを先に考えれば良い．つまり，引き分けを考えなくて済むということである．更に，非整数値 a に対して，$\Delta(a+1)=\Delta(a)+\mathrm{P}[\lceil a+p\rceil]-\mathrm{P}[\lceil a\rceil]$ である．これは $a<$ MinScore $-\,1$ のとき，第 3 項が 0 であり，広義の短調増加であることを意味する．また，MaxScore $-\,p<a$ のとき，第 2 項が 0 となり，広義の短調減少であることを意味する．加えて，自明ではあるが，$a<$ MinScore $-\,p$, MaxScore $<a$ の時，$\Delta(a)=0$ である．従って，$\Delta(a)$ が最大となりえる範囲は MinScore $-\,1<a<$ MaxScore $-\,p+1$ である．

以上から $\lceil a'\rceil=$ MinScore，$n=$ MaxScore $-$ MinScore $-\,p+1$ を満たす非整数 a' と整数 n について，$\Delta(a'),\Delta(a'+1),\cdots,\Delta(a'+n)$ のいずれかが最大値をとる．

非対称ゲーム (一律) 長さ 3 幅 4 の例を挙げて説明する．次善手のペナルティが 1 であるので，以下

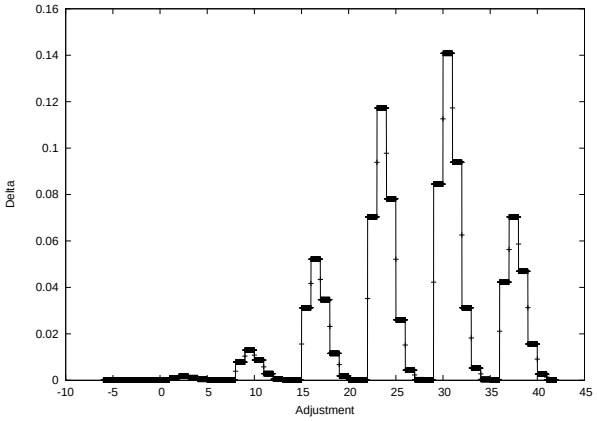

図 7 非対称なゲーム (一律)，盤の長さ 6 幅 4 とした場合の各補正量での最善手と次善手の期待値の差 (Δ) の値

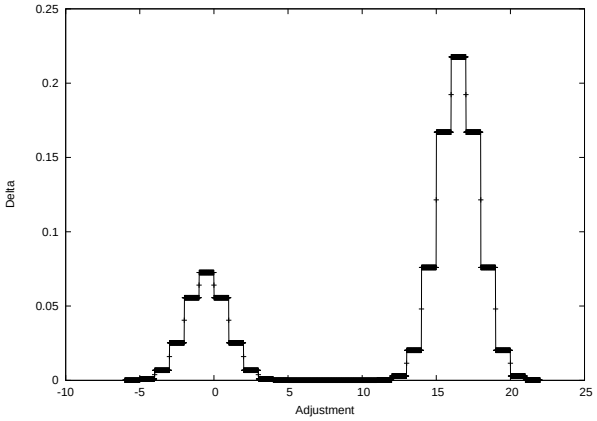

図 8 非対称ゲーム (最終手)，盤の長さ 6 幅 4 とした場合の各補正量での最善手と次善手の期待値の差 (Δ) の値

$p=1$ とする．非整数補正量 a に対して，$\Delta(a)=\sum_{a<s<a+1}\mathrm{P}[s]=\mathrm{P}[\lceil a\rceil]$ であり，これを最大化する．プレイアウト中相手は 3 回中 m 回ペナルティを踏み，UCT プレイヤが 2 回中 n 回ペナルティを踏む場合の確率は $\mathrm{P}[4m-n]={}_3\mathrm{C}_m(\frac{1}{4})^m(\frac{3}{4})^{3-m}{}_2\mathrm{C}_n(\frac{1}{4})^n(\frac{3}{4})^{2-n}$ である．従って，$(m,n)=$ (2,2) と (3,2) が最適値である．それに対応する補正量 a は $9<a<10,5<a<6$ となる．このようにして，本稿で扱うゲームでは任意の長さ，任意の手番の探索の場合の最適補正量を求めることができる．

また，非対称ゲーム (最終手) でも，同様に盤の長さ 6，幅 4，初手探索時に，補正量を変えた場合の Δ の値を 0.01 間隔でプロットした．**図 8** から，補正量 $4<a<12$ で Δ はほぼ 0 であることや，Δ は $16<a<17$ で最大になることが見て取れる．以下簡単のため非整数値補正量 a を $\lceil a\rceil-0.5$ で代表させることにする．例えばこの場合の最適補正量は 16.5 である．

3.2.2 最大頻度法

提案手法 (Algorithm3) は，スコアのヒストグラムを取り，ヒストグラムの最大値のスコアを補正値とする調整法である．Histogram はスコア (補正前) の頻度を管理する配列とする．Adjustment は dynamic komi のコミに相当する補正量であり，囲碁に限定していないので，Adjustment とした．

Algorithm 3 MAX FREQUENCY

Adjustment $\leftarrow \arg\max_{s \in \text{Score}} \text{Histogram}[s]$

この手法の利点は，大きな Δ を持つ補正量が選ばれると期待できることである．直感的には，図 3 および図 6 に描いた頻度のヒストグラムと，図 7 および図 8 に描いた Δ が同じ形で変化していれば，最大頻度の補正量が最適に近い．両者がどの程度一致するかは条件に依存するが，少なくとも Δ の定義である式 (3) から明らかなように，スコア s の頻度が 0 でない限り $\Delta(s)$ は 0 にはならない．すなわち提案する最大頻度法は，選ばれた補正量で Δ が 0 になることはないという保証を持つ．一方，dynamic komi を用いた場合はその保証はない．dynamic komi では期待値付近を補正量として選ぶため，スコアのヒストグラムが多峰の場合に補正量は谷に設定されうる．そのようにして $\Delta = 0$ となった場合は，1 手読みの UCB1 で探索した場合に, プレイアウト数をいくら増やしても最善手とその他を区別できないことを意味する.

最大頻度法のもう一つの利点として，図 3 のようにスコアのヒストグラムが複数の山に分かれている場合に，最善手が識別しやすいことが挙げられる．図 3 ではそれぞれの山の中で最善手とそれ以外の手が少しずれて分布している．一番頻度が多い山に境界を移動することにより, それらを効率的に見分けられると期待できる. さらに最大頻度法は，置き碁のように山が一つで全体がずれている場合でも，補正量を調整可能である．従って, 様々な場面で働く頑健な手法と期待される.

しかしこの方法にも，dynamic komi と同様に補正量が変化することによる悪影響が起こりうる．本来同じスコアには同じ利得を返すべきだが，補正量の移り変わるタイミングによっては同じスコアに違う利得が与えられる．これが違うノードでのプレイアウトで起きると木の成長の仕方が変わることにつながる．ただ，後に述べるように実験では目立った不具合は見られなかった.

4. 実験結果

この節では，各非対称ゲームで対戦実験をして，提案手法と既存手法を用いた場合の勝率を比べる．また，勝率を Δ の大小の観点から論じる.

4.1 非対称ゲーム (一律)

このような難しい探索問題に対して，提案手法の性能を調べるために，対戦実験を行った．提案手法の比較対称として，それぞれ，3.2.1 で定義した最適補正量，dynamic komi の Score Situational(以下 Score と英字で表記) と，Value Situational(Value) に基づく補正と更に補正しない通常の UCT の計 5 つの手法を用いて図 4 と同様に盤の長さを変えた場合の勝率の変化を測定した．尚，dynamic komi のパラメータは文献 [2] のものでは勝率が低かったので，調整し，Score で $s = 0, c = 5$, Value で red $= 0.3$ green $= 0.85$ とした．**図 9** から読み取れるように，勝率の高い順に最適補正 (Theoretical)，提案手法 (MaxFrequency), Value, Score, 通常の UCT(Normal) となった．提案手法は最適補正には及ばなかったが，既存の手法を大きく上回る勝率となった.

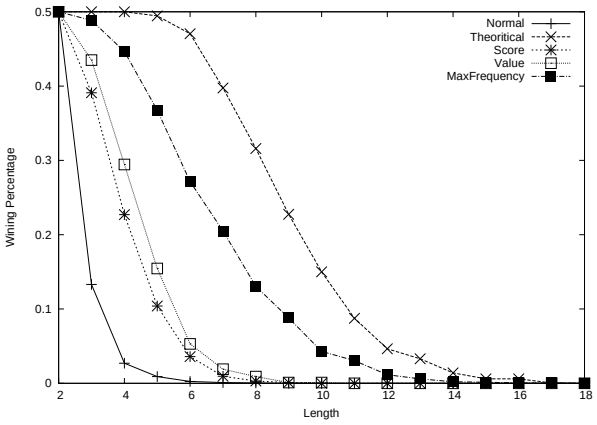

図 9 非対称ゲーム (一律) で盤を長くした場合の各アルゴリズムの勝率の変化

次に，非対称な仮想ゲームでの結果について，特に長さ 6 の盤のある試合での初手探索時のデータを Δ の大きさの観点から分析する．長さ 6 の場合を取り上げたのは，勝率が一番低いものでも勝率の下限 0 より高く，また一番高いものでも勝率の上限 0.5 よりは低いためである．ここでは，プレイアウトを増やした場合の各アルゴリズムでの補正量の変化を見る．**図 10** を図 7 を参照して見比べる．まず，静的な補正法では，補正量 0 の場合 Δ の値は $1.2 \cdot 10^{-7}$，最適補正量 30.5 の場合 0.14 となっている．次に動的な補正法について述べると，Score の場合は，補正量が 10 から 40 ぐらいを揺れ動いていて，Δ が最大になったり，最小になったりしている．平均するとあまり高く無いと言える．但し，補正量 0 の時よりは当然ながら高い．その次に Value の場合は，補正量が 1 ずつ増え 27 で安定している．補正量 27 では Δ は 0.0023 であり，Score での平均した値よりも低い値である．最後に提案手法 (MaxFrequency) の場合は，補正量は最適補正量 ± 0.5 を行ったり来たりから最後は 24 で安定している．補正量が最適補正量 ± 0.5 の時は Δ は 0.11,0.12 であり，24 の時も 0.098 で他の手法と比べて高い値となっている．そのような理由で他と比べて高い勝率を達成できたのだと予想される.

4.2 非対称ゲーム (最終手)

補正量の重要性が分かる例として非対称ゲーム (最終手) での実験結果を示す．図 9 の実験と同様に 5 種の手法で長さを変えた場合の勝率を測定した．尚 Score, Value のパラメータも前節同様の Score で $s = 0, c = 5$, Value で red $= 0.3$, green $= 0.85$ とした．その結果**図 11** に示したようにと勝率の高い順から最適補正 (Theoretical)，提案手法 (MaxFrequency)，通常の UCT(Normal)，dynamic komi となった．dynamic komi の 2 種はほぼ同

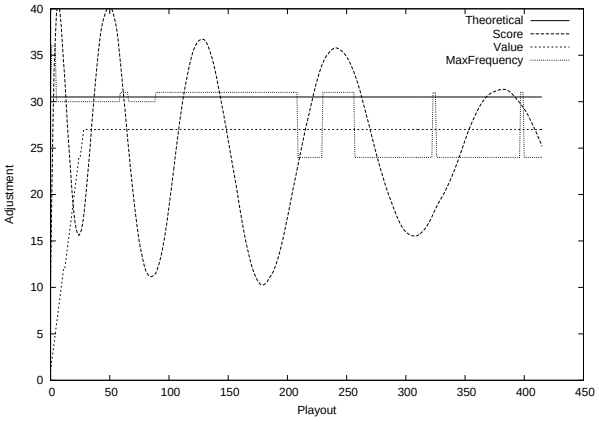

図 10 非対称ゲーム (一律) で各アルゴリズムで毎プレイアウトでの補正量の変化

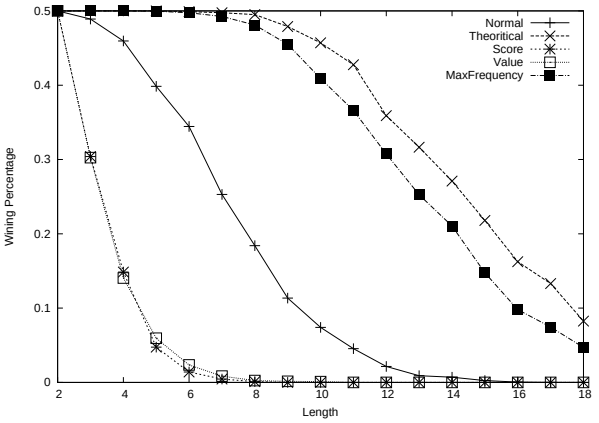

図 11 非対称ゲーム (最終手) で盤を長くした場合の各アルゴリズムの勝率の変化

じ勝率だった．重要な点は通常の UCT よりも dynamic komi は勝率が低くなったことである．一方で dynamic komi が上手く行かない仮想ゲームでも提案手法は高い勝率を達成できた．

前節と同様に盤の長さ 6 の時のデータを分析する．補正量の変化を**図 12** に示す．図 8 と見比べると，提案手法 (MaxFrequency) の補正量は最適補正量 ±0.5 で安定している．補正量 16, 16.5(最適補正量), 17 での Δ の値は，それぞれ 0.19, 0.21, 0.19 となり，提案手法での Δ の値は最適補正量での値に近くなることが分かる．一方で，dynamic komi の 2 種類は，共に Δ が 0 の所に境界を移動させるような補正量となっていることが多い．Score の場合，Δ がほぼ 0 となる範囲 4 から 12 に補正量が設定されて，2/3 近くのプレイアウトが行われている．また，Value の場合補正量 5 で安定しているが，この時 Δ は $2.9 \cdot 10^{-5}$ でほぼ 0 となっている．このゲームでは，補正量が 0 でも Δ の値は 0.064 となり，dynamic komi を使うとこれよりも小さい値となっていて，明らかに dynamic komi が悪影響を及ぼしていると言える．

5. おわりに

本研究では UCT にとって難しい探索問題の性質の一つを明らかにした．具体的には MinMax 値は偏っていないが，各手番で勝ちやすさが異なっていて，相手プレイヤの手番では最善手以外を選ぶと相手の負けが確定するような場合である．そのような状況では UCT で行われるプレイアウトの結果がほとんど勝ちとなり，手の評価に差が付かないため，最善手を選ぶのが難しい．その問題に対して，頻度最大のスコアに境界を移動する提案手法が dynamic komi 等の既存手法を大きく上回る性能であったことを示した．また，dynamic komi が通常の UCT よりも性能が劣る探索問題の性質を示し，提案手法はその状況でも有効である事を示した．さらに，UCT でなく UCB1 で探索することを仮定した場合の，最善手と次善手の利得の期待値の差の大きさの観点から，最適な補正量を求め，各手法の優劣を論じた．

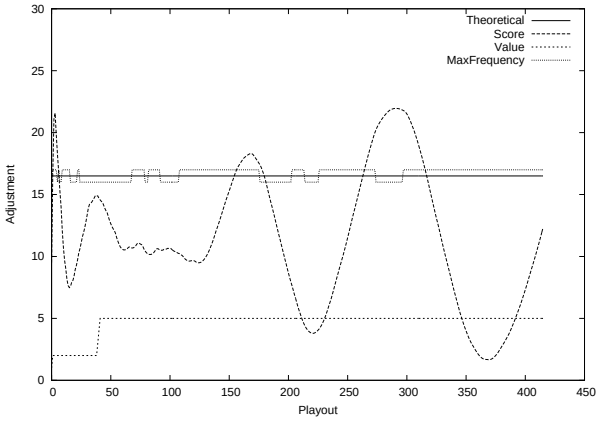

図 12 非対称ゲーム (最終手) で各アルゴリズムで毎プレイアウトでの補正量の変化

提案手法の有効性を囲碁等の複雑なゲームで検証することが今後の課題である．また，補正量を変えるアルゴリズムで，補正量が必要以上に揺れ動くことの悪影響についてはあまり分かっていない．しかし，実際に対戦で効果を挙げるためには影響を見積る研究も必要であると考えられる．

参考文献

[1] Peter Auer, Nicolò Cesa-Bianchi, and Paul Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine Learning*, Vol. 47, No. 2-3, pp. 235–256, 2002.

[2] P. Baudiš. Balancing mcts by dynamically adjusting the komi value. *ICGA Journal-International Computer Games Association*, Vol. 34, No. 3, p. 131, 2011.

[3] Hilmar Finnsson and Yngvi Björnsson. Game-tree properties and mcts performance. In *Proceedings of 2nd International General Game Playing Workshop (GIGA2011)*, pp. 23–30, 2011.

[4] L Kocsis and Cs Szepesvari. Bandit based monte-carlo planning. In *Machine Learning: ECML 2006*, Vol. 4212, pp. 282–293. Springer, 2006.

[5] T. L. Lai and Herbert Robbins. Asymptotically efficient adaptive allocation rules. *Advances in Applied Mathematics*, Vol. 6, No. 1, pp. 4–22, 1985.

[6] Platzner Marco Tobias Graf, Lars Schaefers. On semeai detection in monte-carlo go. In *The 8th International Conference On Computers And Games*, 2013.