修　士　論　文

コンピュータゲームプレイヤにおける勝率を用いた探索時間最適化手法

Optimization of search time with winning percentages for computer game players

指導教員　**近山 隆 教授**

東京大学大学院新領域創成科学研究科
基盤情報学専攻

氏 名　**56301 阿部 崇史**

提 出 日　**平成 19 年 1 月 31 日**

概 要

現在のコンピュータゲームプレイヤは１手あたりの持ち時間を予め決められた探索の深さ、または予め定められた時間制限によって決定しているケースが多い。そのため、１手に使う持ち時間にあまり大きな違いが見られない。ところが、人間の思考時間はそうではない。人間は、局面の状況から長く考えるべきか、それとも短く考えるべきかを判断し、自分の持ち時間を上手に配分している。本研究では、コンピュータゲームプレイヤに持ち時間の適切な配分をさせることを目標としている。そこで、２つの段階に分けて探索時間を調整しようと考えた。１段階目は、長考すべきかどうかを判断するマクロな視点からの調整である。２段階目は、特定の局面において勝率を用いて適切な探索時間を求めるミクロな視点からの調整である。これらの視点から探索時間の最適化を行うために、反復深化中の最善手の変化、探索時間と評価値の関係、および評価値と勝率の関係を調べ、探索時間を一手毎に最適に割り振るコンピュータゲームプレイヤを作ることを試みた。本研究ではオリジナルのプレイヤと強さが変わらない結果となった。長く考えるべき局面の発見の仕方や一手当たりの探索時間の最適化の仕方などは今後更なる研究の余地があることが分かった。

目 次

第 1 章　序論 **1**
1.1　背景と目的 2
1.1.1　背景 2
1.1.2　目的 3
1.2　本論文の構成 4

第 2 章　関連研究 **5**
2.1　評価関数 6
2.2　ゲーム木探索 7
2.2.1　代表的な探索手法 7
2.2.2　実現確率打ち切り探索 9
2.3　探索の深さと探索時間の関係 11
2.3.1　探索の深さと探索時間の関係 11
2.3.2　探索時間と実現確率の閾値の関係 11
2.4　反復深化 13

第 3 章　勝率を用いた探索時間の最適化 **15**
3.1　提案手法 16
3.1.1　勝率の最大化 16
3.1.2　探索時間の最適化における問題点 16
3.2　基礎実験データ 18
3.2.1　探索時間と評価値の関係 18
3.2.2　探索時間と勝率の関係 19
3.2.3　評価値と勝率の関係 23

第 4 章　実装 **29**
4.1　実装方法 30
4.2　実験とその結果 32
4.3　考察 35

第 5 章　まとめと今後の課題 **38**
5.1　まとめ 39
5.2　今後の課題 41

図 目 次

2.1 min-max 探索木 . . . 7
2.2 実現確率打ち切り探索木 . . . 9
2.3 閾値と探索時間の関係 . . . 12
2.4 反復深化 . . . 13

3.1 探索時間と評価値のぶれの関係 . . . 18
3.2 深さの差と勝率の関係（深さ 16） . . . 20
3.3 深さの差と勝率の関係（深さ 15） . . . 20
3.4 深さの差と勝率の関係（深さ 14） . . . 21
3.5 深さの差と勝率の関係（深さ 13） . . . 21
3.6 評価値と勝率の関係（進行度 32） . . . 25
3.7 評価値と勝率の関係（進行度 64） . . . 26
3.8 評価値と勝率の関係（進行度 96） . . . 26
3.9 進行度による関係式の変化 . . . 27

4.1 深さ 16 での対局時の探索時間 . . . 33
4.2 深さ 15 での対局時の探索時間 . . . 33
4.3 深さ 14 での対局時の探索時間 . . . 34

表 目 次

3.1 近似式 22
3.2 進行度と勝率関数の係数 28

4.1 提案手法を実装したプレイヤの勝率 32

第１章　序論

1.1 背景と目的

1.1.1 背景

コンピュータゲームプレイヤは、人工知能の一つの分野として研究されてきた。コンピュータ開発の黎明期であった 1950 年には既に Claude E.Shannon や Alan Turing といった研究者によってチェスプログラムの論文が発表されており、1950 年代の内に最初のチェスプレイヤが作られた。その後、コンピュータの性能の向上と共にコンピュータゲームプレイヤの強さも向上し、1997 年にはついにチェスの世界チャンピオンを破ることに成功する。また、同年に行われたオセロでの世界チャンピオンとの戦いでは 6 番勝負を行い全勝するという結果を残した。チェスやオセロの世界では、既にコンピュータが人間に追いついているのである。

近年、コンピュータゲームプレイヤの開発は、将棋や囲碁といった探索領域の広いゲームに関心が移ってきている。今回研究の対象とするのは将棋のプレイヤである。オセロの探索領域は約 10^{60}、チェスの探索領域は約 10^{120} 程度であり、人間が直感的に手を絞って読むよりも、コンピュータがしらみつぶしに手を読む方が有効になる探索範囲になってしまった。それに対し、将棋は持ち駒を打てることが出来るために探索領域は大きく広がって約 10^{220}、囲碁に至ってはその盤面の広さにより約 10^{360} の探索領域があり、人間の直感で手を絞る方がコンピュータの探索よりも有効なのである。そのため、序盤は定跡を利用する、手の選択肢の多い中盤では有効だと思われる手に絞るといった heuristic な探索が必要となる。

しかしながら、コンピュータ将棋プレイヤに heuristic な探索を行わせることは非常に難しい。盤面の評価値を返す静的評価関数と呼ばれるものには少なからずエラーが入っており、途中まで探索したところでそれが信用にたるものかどうか分からないからだ。将棋の場合、取った駒が再び使えるために駒の価値だけでは正確な静的評価関数を作ることは難しく、このこともコンピュータ将棋プレイヤの成長の妨げの一因となっている。

終盤に限って言えば、コンピュータは人間を越えているという見方もある。終盤の攻防である寄せは序盤、中盤に比べて極めて手が限られているために、見た目よりも探索領域は小さく、数十手先まで読み切れてしまうのである。詰め将棋だけであれば、現在最長とされている 1525 手詰めの『ミクロコスモス』という問題を数時間で解いてしまうまでに至っている。現時点で最強のコンピュータ将棋プレイヤの強さはアマチュア 4 段程度とされているが、それはこの終盤での強さも含めての評価であり、序盤から中盤にかけてはまだまだアマチュアレベルの棋士にも勝てていないのである。[4]

1.1.2 目的

本研究室ではコンピュータ将棋プレイヤの研究を行っている。現在のコンピュータ将棋プレイヤは１手あたりの持ち時間を予め決められた探索の深さ、または予め定められた時間制限によって決定しているケースが多い。そのため、１手に使う持ち時間にあまり大きな違いが見られない。ところが、人間の棋士の指し方はそうではない。人間は、局面の状況から長く考えるべきか、それとも短く考えるべきかを判断し、自分の持ち時間を上手に配分している。それならば、コンピュータのプレイヤにも人間のように上手に１手に使う持ち時間を配分させることができれば、より強いプレイヤが出来上がるのではないかと考えた。

では、どのようにコンピュータに持ち時間を配分させれば良いのか。長考のメリットは、その局面の評価値がより正確になるということである。これにより、予想外の手で形勢が悪くなる可能性が低くなる。　それに対し、長考することのデメリットは、残り時間が少なくなるので以後の手にかけられる時間が少なくなり、悪手を打つ可能性が増えてしまうことである。　また、長考するかどうかの判断は、長考する前にできるだけ正確に行わなくてはならない。長考してからその長考が意味のないものだったという事態を避けるためである。

コンピュータのプレイヤにこれらのことをさせるためには、数値化を行う必要がある。コンピュータのプレイヤに大量の過去の棋譜を元に考える時間の長さを学習させるのが一番単純な方法である。しかし、このやり方では１局面毎にプレイヤに探索を行わせる必要があり、学習に非常に時間がかかると予想され、現実的ではない。そこで、本研究では２つの段階に分けて探索時間を調整しようと考えた。１段階目は、長考すべきかどうかを判断する段階である。これを判断させるために、探索の途中でどれだけ最善手が変化するかを調べた。これは、最善手の変化が多い局面は候補手が多いため、探索に多くの時間が必要となる可能性が高いと考えたからである。２段階目は、適切な探索時間を求める段階である。これを求めるために、探索時間と評価値の関係、および評価値と勝率の関係を求め、それらを組み合わせることによって局面から適切な探索時間を求めることができないかと考えた。

1.2 本論文の構成

以降、本論文の構成は次のようになっている。

第 2 章 関連研究

ゲーム木探索、進行度と勝率の関係といった関連研究について述べる。

第 3 章 勝率を用いた探索時間の最適化

本研究で行った勝率を用いた探索時間の最適化について述べる。

第 4 章 実装

本研究で提案した手法を実装した過程と結果について述べる。

第 5 章 まとめと今後の課題

本研究のまとめと今後の課題について述べる。

第 2 章　関連研究

本章では関連研究として、2.1 においてコンピュータゲームプレイヤを作成する上で欠かすことのできない静的評価関数について説明する。次に 2.2 において、多くのコンピュータゲームプレイヤが用いている探索手法と本研究で用いたコンピュータゲームプレイヤが用いている探索手法である実現確率打ち切り探索について紹介する。次に 2.3 において、探索の深さと探索時間の関係について説明し現在のコンピュータゲームプレイヤが抱えている探索にかける時間についての問題点を挙げる。さらに 2.4 において、多くのコンピュータゲームプレイヤで取り入れられている反復深化という探索を効率化する手法について説明する。

2.1 評価関数

将棋のような探索空間の大きいゲームでは現実的な時間内で手を全て探索することは不可能である。そのため、探索を途中で一度停止し、探索できた局面での有利不利を元に、自分が勝ちに近づいているかどうかを判断しなければならない。評価関数とは、探索木のそれぞれの局面の有利不利を数値として評価する関数のことである。一般的にプラスの値が大きいほど味方有利、マイナスの値が大きいほど味方不利で、値がゼロであれば互角であるとする。

評価関数はヒューリスティックな関数である。コンピュータゲームプレイヤの評価関数の場合、プレイヤを作ろうとしているゲーム特有の定跡や有効な駒の形などをこの評価関数で判断させることになる。もし仮に絶対の評価基準が存在した場合、コンピュータゲームプレイヤは探索を行わずに評価関数の出した値のみを元にゲームをプレイすれば確実に勝つことができる。将棋やチェスといった完全情報ゼロ和ゲームは、勝敗がつく以上可能な全ての駒の動かし方を調べれば先手必勝か後手必勝に辿り着いてしまう。しかし評価関数にそれを記述することは不可能である。つまり、評価関数にはどこかしら必ずエラーが含まれているということになる。評価関数には作成者の知識や経験による評価基準が存在し、これがエラーとなっているのである。

当然のことながら、この評価関数が正確であればあるほどコンピュータゲームプレイヤは強くなる。そのため、評価関数の中に多くの評価要素を詰め込んで局面の正確な評価ができるようにするのである。しかし、ゲーム木探索では 1 秒間に数万局面以上の非常に多くの局面を評価することになるため、1 局面の評価に時間をかけるわけにはいかない。そのため評価関数は十分に速くなくてはならない。しかし、正確な静的評価関数を作ろうとすると評価要素は多くない遅くなってしまう。つまり評価関数の速度と正確さはトレードオフの関係にあり、正確かつ高速な評価関数を作ることは非常に困難なのである。

2.2 ゲーム木探索

2.2.1 代表的な探索手法

将棋は各プレイヤが交互に手を指すゲームである。そこで、現在の局面を表すものをを作る。これをノードという。1 手指す毎にそのノードの下に新しい局面を表すノードを作るという作業を繰り返すと、対局の流れを、局面同士が手で繋がれた木構造で表すことが出来る。この時、各プレイヤが交互に手を指しているので、お互いの局面が交互に存在する形になる。こうして作られた木の中で、最も適当だと思われる最初の一手を見つけることが探索の目的である。現在のコンピュータゲームプレイヤには min-max 法という探索手法多くが用いられている。今までに min-max 法以外の探索方法も考えられてきたが、結局は min-max 法の方が強いプレイヤが作れることが多いため、min-max 法を改良した探索手法が使われているのである。

以下に min-max 法を説明を述べる。min-max 法では各 node の値を探索で求めた後、自分は最も評価値の高い node を選択し、相手は最も評価値の低い node を選択すると仮定して手を選択する。図 2.1 の探索木では各 node に評価値が与えられている。この評価値を決定するのには静的評価関数を用いる。静的評価関数とは読みを行わずに形勢を評価し局面の点数を決定する関数である。例えば将棋では、駒の位置、駒の利き、自分の持ち駒、玉や大駒の自由度といった特徴量を基にして点数を決定し、盤面の有利不利を数値化することにしている。特に記述が無い場合、点数の高い方が自分に有利な局面であるとする。

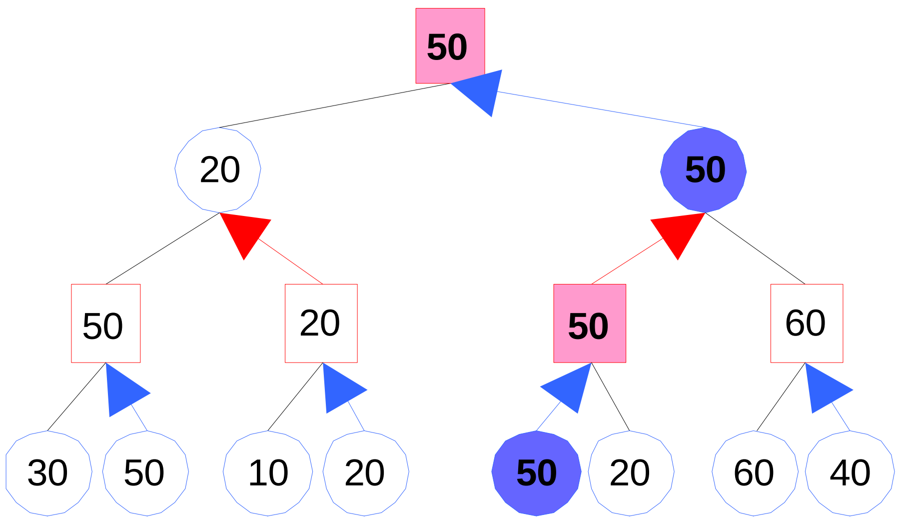

図 2.1: min-max 探索木

□が自分の手番のノード (max ノード) を表し、○が相手の手番のノード (min ノード) を表す。ノードに書かれている数字は、後述する評価関数によって求められたその局面の評価値である。木の末端のノードのことを葉ノードと呼ぶ。ノードの展開が終了した時点で評価関数を呼び出し、葉ノードに対して評価値を定める。min-max 探索において、最も適当な手は次の規則によって選ばれる。

- max ノードでは子ノード (自分の下にあるノード) のうち最も大きい値のノードを選択する (自分は自分にとって最も有利な手を選択する)
- min ノードでは子ノードのうち最も小さい値のノードを選択する (相手は自分にとって最も不利な手を選択する)

評価値の与えられた葉ノードからこの規則にしたがって親ノード (自分の上にあるノード) を選んでいき、最終的に root ノード (現在の局面を表すノード) に到達した時点で選ばれた手が最も適当な手であると判断する。[2]

2.2.2 実現確率打ち切り探索

実現確率打ち切りとは、深さの代わりに実現確率の積を用いたものである。実現確率というのは、その局面に至るまでの手順が実際に指されて、その局面が実現する確率のことである。実現確率打ち切りでは、手に対して遷移確率を設定し、遷移確率の積、即ちノードの持つ実現確率が閾値を下回った時点で探索を中断する。例えば、ある指し手 M によって局面 A が局面 A' に変化するとき、局面 A' の実現確率は次のように計算される。

$$\text{局面 A' の実現確率} = \text{局面 A の実現確率} * \text{指し手 M が指される確率} \tag{2.1}$$

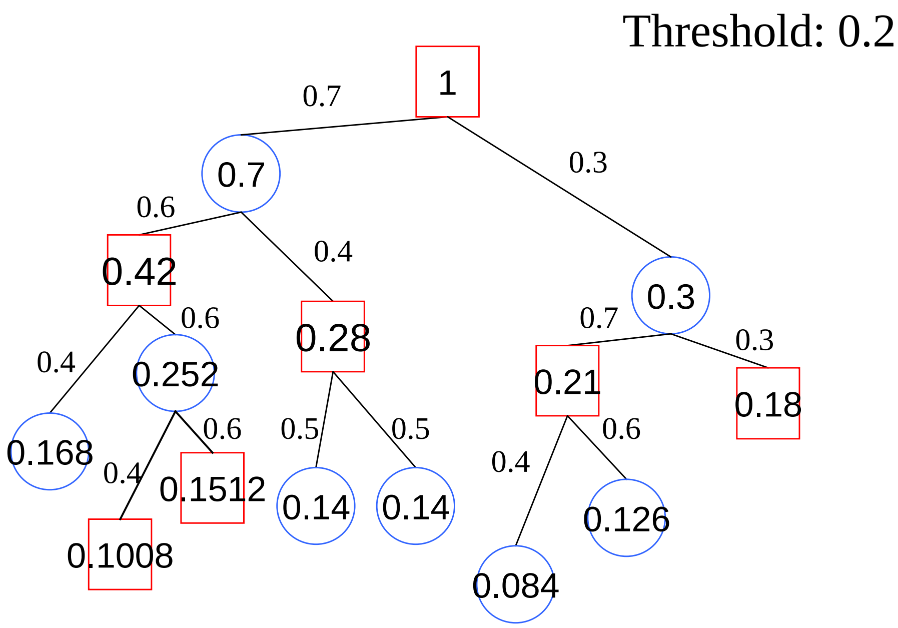

図 2.2: 実現確率打ち切り探索木

ルート局面は既に実現しているので実現確率は 1 である。読みの中での局面の実現確率は、1 から始まって読みが深くなるにつれて徐々に減少していくことになる。その実現確率が閾値を下回った時点でノードの展開を中断する。このようなアルゴリズムを用いることによって、実現する確率の高い「ありそうな局面」のみを読むことができる。この手法では、ある局面が次の局面に移る時の遷移確率をどう求めるかが問題になる。本研究室で扱っているコンピュータ将棋プレイヤ『激指』ではプロ棋士の実戦棋譜約 600 局から統計情報を抽出

し、遷移確率の計算を行っている

ここで、実現確率の計算が掛け算であることに注目する。実現確率に対して log を取ると、実現確率の積は全て log の足し算で表すことができると分かる。(激指ではこれを log p と表している) つまり実現確率打ち切りは普通の min-max 探索と同じアルゴリズムで、深さの増加分を指し手の種類に応じて可変にしただけということになる。[3]

2.3 探索の深さと探索時間の関係

2.3.1 探索の深さと探索時間の関係

通常のコンピュータゲームプレイヤが行っている探索は min-max 探索である。この min-max 探索を用いている場合、探索の深さ d と探索する葉ノードの数 $leaves$ は平均分枝数 b を用いて以下の様な式で表すことができる。

$$leaves = b^d \tag{2.2}$$

この式を見ると分かるように、探索を行うノードの数は指数関数的に増加していく。そのため、d が非常に大きくなった場合、1 手先まで読む、つまり d が 1 増えるだけで、探索しなければならない葉ノードは急激に増加してしまい、現実的な時間内で探索を終えることが不可能になってしまうのである。そのため通常の min-max 探索では、探索の深さを変えることによって探索時間を調整することは非常に難しい。[5]

反復深化しながら探索を行っていた場合、探索をしている途中で制限時間が来てしまうと最後に探索した部分は無駄になってしまい、その前の反復深化までの結果しか使えないことになってしまう。[6] この問題は深く読めば読むほど起こりやすく、時間を有効に使うことの大きな妨げとなっている。実現確率打ち切りではこの問題を実現確率の閾値を調整することによって解決することができる。実現確率の閾値は深さのような飛び飛びの値とは違い、連続で変化させることができる。そこで、残り時間が少なくなった時には残り時間を丁度使いきれるような実現確率を閾値に設定できれば良い。

2.3.2 探索時間と実現確率の閾値の関係

2.2.2 において説明した実現確率の閾値を設定するためには探索時間と実現確率の関係が必要である。そこで実現確率の閾値を変化させた時にどれだけ探索時間が変化しているのかを調べた結果を以下に示す。用いたコンピュータゲームプレイヤは本研究室で扱っているコンピュータ将棋プレイヤ『激指』である。

2.3 を見ると、実現確率の閾値を 2 だけ変化させると必要な探索時間は約 6 倍となっていることが見て取れる。激指の実現確率 $logp$ と探索時間 $time$ の関係式は

$$time = 0.037 \exp^{0.9153 logp} \tag{2.3}$$

という式で近似される。この関係式を用いて実際の探索時間を調整する。

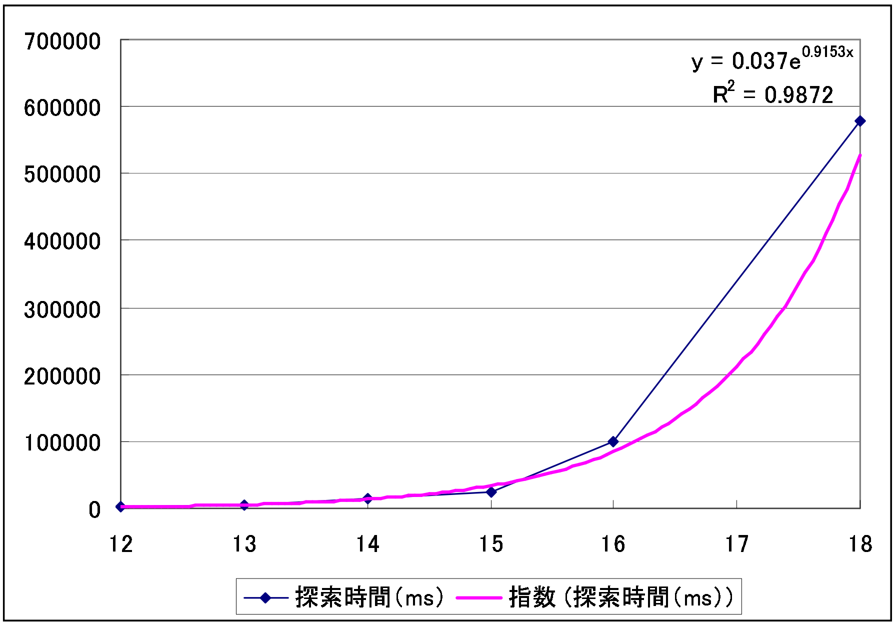

図 2.3: 閾値と探索時間の関係

これ以降、本論分ではプレイヤの実現確率の閾値のことを簡単に『深さ』と表現することにする。これは、2.2.2 で述べたように、実現確率の閾値は通常の min-max 探索の深さに相当するからである。

2.4 反復深化

2.2 で紹介したゲーム木探索では、末端まで探索が終了しなければ root ノードの評価を出すことができない。そのため、制限時間内に一度に深く探索しようとすると探索しきることができずに探索結果が出ないという事態に陥ってしまう可能性がある。反復深化（Iterative deepening）とは、このような事態を避けるためにだんだんと探索する深さを深くしていくという探索アルゴリズムの改良手法である。

反復深化では、まず一度に深くまで探索を行わずに浅いところから探索を始める。そして徐々に探索の深さを深くしていく。こうすることによって、制限時間内で探索可能な深さまで探索することができる。また、反復深化は枝刈り（pruning）という手法とも相性が良い。枝刈りも多くのコンピュータゲームプレイヤで用いられている探索量を減らす探索アルゴリズムの改良手法であるが、ここでの詳しい説明は割愛する。

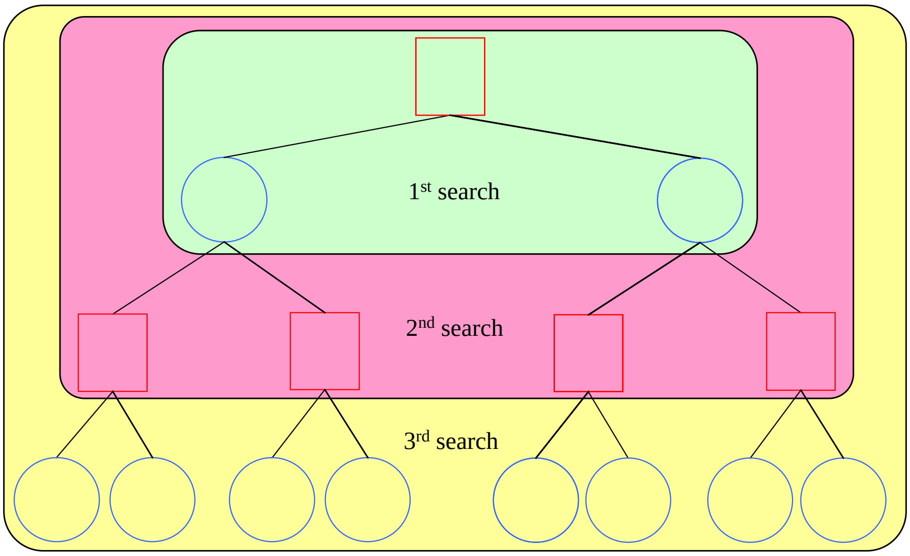

図 2.4: 反復深化

反復深化を行うと浅い部分のノードは何度も探索を行うことになるため探索量が増加する。平均分枝数を b とおくとある深さ k まで探索する時の探索ノード数 sum は

$$sum = 1 + b + b^2 + \ldots + b^k \quad (2.4)$$

$$= \frac{b^{k+1} - 1}{b - 1} \quad (2.5)$$

となる。よって深さ 1 ごとに深さ d まで反復深化を行ったときの探索ノード数 sum_{id} は

$$sum_{id} = \sum_{k=0}^{d} \frac{b^{k+1}-1}{b-1} \tag{2.6}$$

となり、d が大きくなると探索ノード数が一度で全て探索する場合に比べてかなり増加することになってしまう。探索する深さを 1 ずつ増やすのではなく一度にもっと深くまで探索するようにすれば反復深化による探索のコストは減少するが、その分枝刈りによる探索の効率化や細かい探索の深さの調整が難しくなる。そのため、多くのコンピュータゲームプレイヤは再び自分の手に戻ってくる 2 手毎に探索する深さを増やしているケースが多く見られる。[1]

第 3 章　勝率を用いた探索時間の最適化

本章では、本研究でのメインテーマとなる勝率を用いた探索時間の最適化について述べる。まず、3.1 において、本研究で提案する探索時間の最適化手法を説明する。

次に、3.2 において、本研究を進める上で必要なデータを取るために行った実験とそのデータについて説明する。3.2.1 では、探索時間を長くすることにより読みが正確になりプレイヤがより強くなることを説明する。3.2.2 では、探索時間を変化させることでどの程度勝率が変化するかを説明する。3.2.3 では、局面の進行度別に評価値と勝率にどのような関係があるかを説明する。

3.1 提案手法

3.1.1 勝率の最大化

人間が持ち時間を一手毎に配分する際には、自分の経験やその時の局面などの様々な要因を元に感覚的に思考時間を決定していると考えられる。しかし、コンピュータゲームプレイヤに感覚で持ち時間を配分させることはできない。そのためコンピュータゲームプレイヤに持ち時間を配分させるには、人間が感覚で行っていることを数値化して入力してやることが必要になってくる。

現在のコンピュータゲームプレイヤは評価関数によって局面の有利不利を評価値と呼ばれる値に数値化している。最初に考案したのはこの評価値を最大化するように探索時間を設定してやる手法であった。しかし、探索時間を長くすることは評価値が大きくなることには繋がらないことが分かった。（これについては 3.2.1 にて後述する。）そこで新しく考案したのが勝率を最大化するという手法である。持ち時間に制限が無い場合、探索時間の長いプレイヤの方が勝率が高いことは経験的に知られている。それならば探索時間と勝率の関係、評価値と勝率の関係を求め、双方に共通している勝率を最大化することによって最適な探索時間を求めることができるのではないかと考えた。

3.1.2 探索時間の最適化における問題点

探索時間を最適化には二つのレイヤーが存在する。それは、

1. 対局中のどの局面で長く考える必要があるか
2. ある局面でどれだけ長く考える必要があるか

である。前者は対局中のどの局面で長く考えれば効率よく時間を使えるかというマクロな視点での探索時間の最適化となっている。それに対し後者はある特定の局面においてどれくら

いの時間を使って考えれば良いかというミクロな視点での探索時間の最適化になっている。

3.1.2.1 マクロな視点での探索時間の最適化

『対局中のどの局面で長く考える必要があるか』は一つの対局の中のどの局面で長く考える必要があるかを判断しなければならない。対局が終わってからであれば判断することは比較的容易であるが、対局中に『ここで長く考える必要がある』と判断するのは容易ではない。

では、コンピュータのプレイヤにそれを判断させるにはどのようにすればよいのだろうか。最も単純な方法は、人間のプレイヤが長く考えた局面をコンピュータのプレイヤに学習させることで長く考えるべき局面の特徴を取り出すことである。しかし今回は、人間のプレイヤが手を指すのにかかった時間を記録してある棋譜を得ることができなかったため、局面の特徴を学習させる方法を取ることができなかった。

そこでコンピュータのプレイヤだけで判断する方法として、反復深化中での最善手の変化を元に長く考えるべきかどうかを判断するという手法を考案した。これは、反復深化の各深さでの最善手の変化を比較するというものである。探索を徐々に深くした時に最善手が変化するということは評価値の前後する手が複数あるということである。このような局面は他の局面に比べて、探索の深さを深くして評価値が前後しなくなるようにした時の効果が高いと考えられる。

3.1.2.2 ミクロな視点での探索時間の最適化

マクロな視点での探索時間の最適化に対して、ミクロな視点での探索時間の最適化は行うべきことがはっきりしている。それは現在の局面でどれだけ長く考えれば良いかを求めれば良いか、といことである。ミクロな視点では、勝率や評価値といった具体的な数値を元に最適な一手辺りの探索時間を求めていくことになる。

本研究は、このミクロな視点での探索時間の最適化において勝率を用いることによって最適な探索時間を求めようと考案されたものである。ある局面で探索時間を長くすると言うことは、次の二つの事項を考えなければならない。

1. 探索時間を長くすることで現在の局面での勝率がどれだけ上がるか
2. 探索時間を長くすることで残り時間が減りその局面以降どれだけ勝率が下がるか

この二つの事項はトレードオフの関係にあり、長く考えれば現在の局面では間違った手を指す可能性が低くなるが、残り時間が少なくなるためにそれ以降の局面で間違った手を指す可能性が高くなってしまう。そこで、探索時間を長くすることによる勝率の上昇と残り時間が

少なくなることによる勝率の低下の値を計算し、勝率が最大となるような探索時間を設定すればそれが最適な探索時間になるのではないかと考えた。

3.2 基礎実験データ

3.2.1 探索時間と評価値の関係

そもそも探索時間を長くすることによってコンピュータゲームプレイヤは強くなるのだろうか、という問題がある。これを証明するために、まず長時間探索を行うことで本当に評価値が正確になっていることを調べた。評価値が正確になっているということは局面を正しく判断できているということであり、より強いプレイヤになっていると考えられるからである。

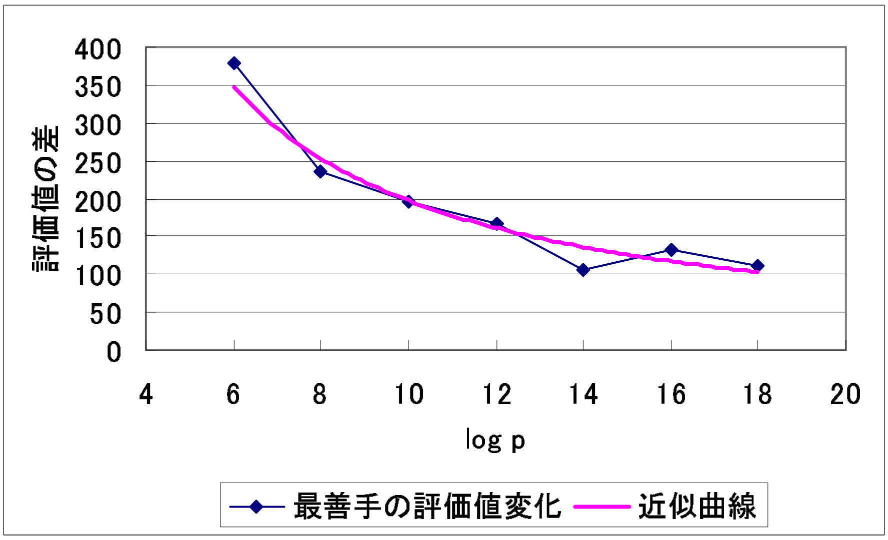

図 3.1: 探索時間と評価値のぶれの関係

グラフの横軸の値は、本研究で用いたコンピュータ将棋プレイヤが探索の際に使っている深さの値である。ここで探索の深さを用いたのは、探索時間を一定にして探索を行うと 2.4 で述べた探索の途中での打ち切りが多く発生してしまう可能性があるからである。また、探索時間と探索の深さの関係は 2.3.2 において示されており、探索を深くすると言うことは探索時間を伸ばすことにあたるからである。縦軸の値は、深さ d を d をから $d + \Delta d$ にしたときに、変化した評価値の絶対値である。評価値の変化はコンピュータ将棋プレイヤの指し手

約 250 手の平均値を用いた。探索を深くする前と後で評価値にずれがあるということは、深くする前の探索による評価が正確でなかったということであり、このずれが少ないほど正確な評価がされていると考えることができる。今回の実験は深さを 2 ずつ増やしていった。グラフを見ると、探索の深さを深くしていくにつれて前の深さとの評価値のずれが少なくなっていることが分かる。これにより、探索を深くすることのメリットである評価値がより正確になるというのは正しいと推測される。

しかし局面の評価値が正確になることは必ずしも局面の評価値をよくするわけではないことが分かった。この実験では、評価値のぶれは小さくなったが浅い読みの時よりも深い読みの時の方が評価値が悪くなるケースも多く見られた。これは、読みが正確になるにつれて自分が不利な状況であることが確実になってきたと判断したからであると考えられる。つまり、探索時間を長くすると言うことはあくまで間違った手を指し難くなるということであって、評価値を上げる手を見つけられるということではないということが分かった。

3.2.2 探索時間と勝率の関係

3.2.1 において探索時間を長くすることで局面の評価値が正確になることを述べた。また同時に、探索時間を長くしても評価値が上がるわけではないことを述べた。従って探索時間と評価値は直接対応させることはできないが、探索時間以外の条件が同じであれば探索時間が長いプレイヤの方が強いことは経験的に知られている。そこで、探索時間と勝率の間にはどのような関係があるかを実験により調査した。以下に行った実験の詳細とその結果を示す。

実験では、一方のプレイヤの探索の深さを固定し他方のプレイヤの探索の深さを変化させることで探索時間を変化させ、探索時間と勝率の関係を求めた。固定する側のプレイヤの探索の深さも変化させ、様々な深さにおける探索時間と勝率の関係を求めた。ここでも 3.2.1 で行った実験と同じように、探索時間の値そのものを変化させるのではなく探索の深さを変化させることによって間接的に探索時間を伸ばして実験を行った。

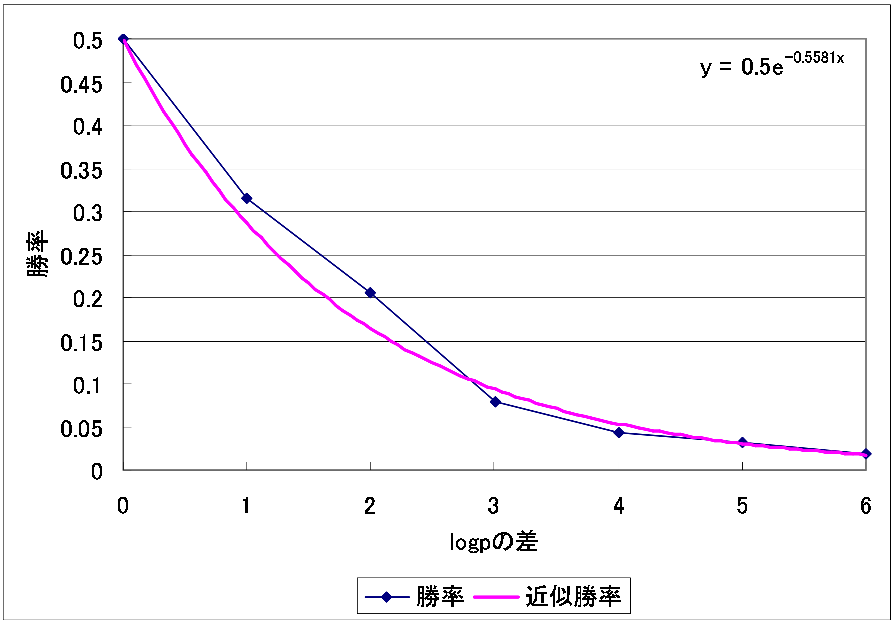

図 3.2: 深さの差と勝率の関係（深さ 16）

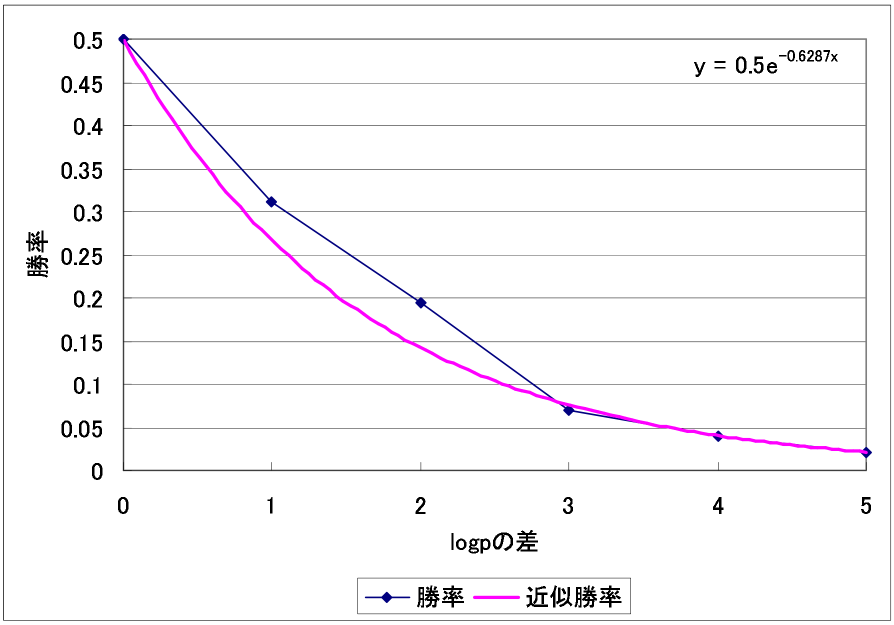

図 3.3: 深さの差と勝率の関係（深さ 15）

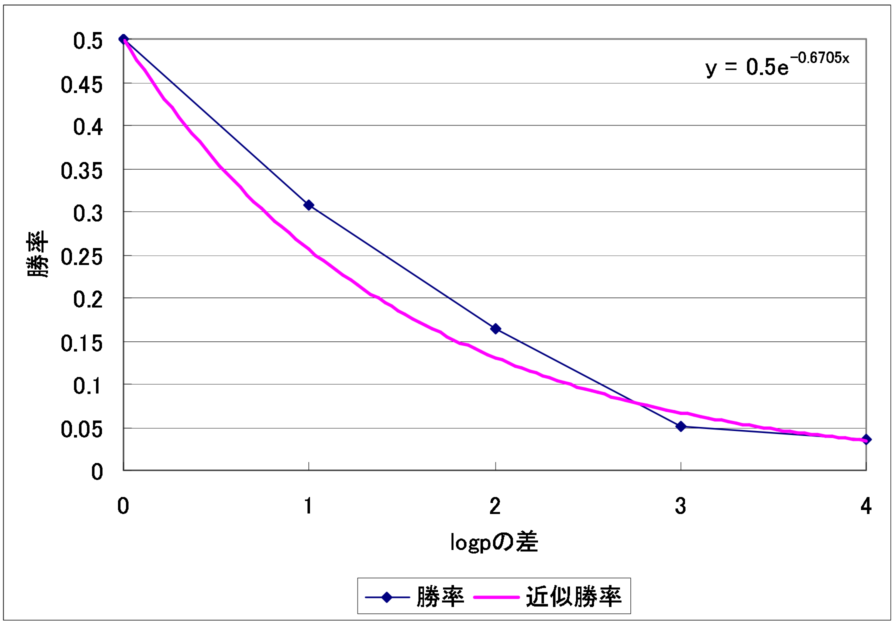

図 3.4: 深さの差と勝率の関係（深さ 14）

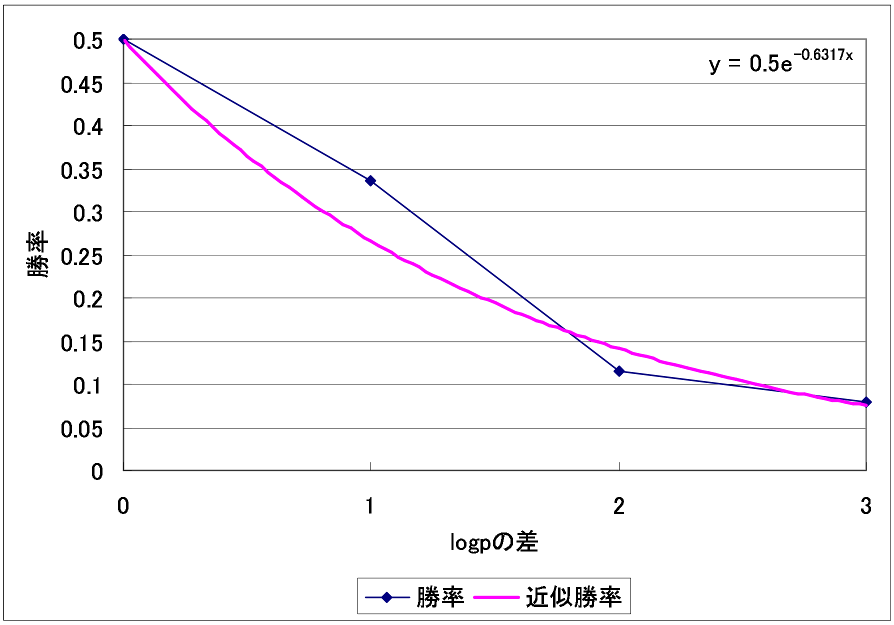

図 3.5: 深さの差と勝率の関係（深さ 13）

行った実験の詳細は以下のようになる。

- 基準となるプレイヤ
  - 探索の深さを 16 から 13 まで 1 ずつ変化させた
- 対局させるプレイヤ
  - 探索の深さを基準となるプレイヤの深さから 1 ずつ減少させていった
- 対局数
  - 500 局

結果は図 3.2〜図 3.5 に示す。これらのグラフを見ると、探索する深さを減少させていくにつれ勝率が指数関数的に減少していくことが分かる。これは例えば、深さ 16 のプレイヤと深さ 15 のプレイヤで対局を行って深さ 16 のプレイヤが勝った試合は深さ 15 のプレイヤを深さ 14 のプレイヤに変更して対局を行っても深さ 16 のプレイヤが勝つと考えられるため、深さ 14 のプレイヤが深さ 16 のプレイヤに勝つ確率 $p_{14,16}$ は深さ 15 のプレイヤが深さ 16 のプレイヤに勝つ確率 $p_{15,16}$ と深さ 14 のプレイヤが深さ 15 のプレイヤに勝つ確率 $p_{14,15}$ を用いて

$$p_{14,16} = p_{15,16} * p_{14,15} \tag{3.1}$$

と表されると考えられる。これは深さ 15 のプレイヤが深さ 16 のプレイヤに勝った対局のうち、深さ 14 のプレイヤが深さ 15 のプレイヤに勝つ対局の存在する確率を表している。このように、深さを変化させた時の勝率の変化は勝率の積で表されると予想されることから、今回の実験結果のグラフに対し指数関数による近似を行った。深さの差を δd とすると、近似の結果は以下のようになった。

基準プレイヤの深さ	近似式
16	$0.5\exp^{-0.5581\Delta d}$
15	$0.5\exp^{-0.6287\Delta d}$
14	$0.5\exp^{-0.6705\Delta d}$
13	$0.5\exp^{-0.6317\Delta d}$

表 3.1: 近似式

深さ 16〜14 までの近似式を見ると、基準となるプレイヤの探索の深さが浅くなるほど深さの差 1 に対しての勝率の変化量が大きくなっていることが分かる。これは探索の深さが浅いほど探索の深さが探索結果に与える影響が大きいことが原因であると考えられる。ところが、深さ 13 のプレイヤを基準としたときには勝率の変化量が小さくなっている。これは探索の深さを変化させるプレイヤの下限を深さ 10 としてしまったために十分な量のデータを取れなかったことが原因であると考えられる。少なくとも深さ 10 程度の深さの探索を行わなければ千日手に陥ったり無駄に駒を行き来したりするだけの手を打ったりしてしまい、有効なデータが取れないと判断したために深さの下限を 10 としたことが裏目に出てしまった結果だと考えられる。

ここで求めた近似式は、4 で激指に実装した手法の中で、延長する探索時間を計算するのに用いた。

3.2.3 評価値と勝率の関係

3.1.1 で述べたように、本研究では勝率を最大化するという手法をとる。勝率を最大化するためには探索時間による勝率の変化の他に、探索時間に依存していない現在の局面の勝率を求められなければならない。現在の局面の勝率が存在し、その勝率を探索時間を変化させることによって最大化するという手法で適切な探索時間を求めようとしているからである。お互いのプレイヤが同じ強さであると仮定すれば、現在の局面が有利な方が勝率が高いと考えられる。そこで、現在の局面の評価値を用いてお互いが同じ深さで探索を行った場合の勝率の変化について実験を行った。

3.2.3.1 局面の進行度

現在の局面の評価値を求めるに当たって考慮しなければならないのが局面の進行度である。進行度とは、現在の局面が対局の中で始めの方にあるのか終わりの方にあるのかを判断する数値のことである。本研究で用いた『激指』では、現在の局面における駒の配置や手数から対局の開始を進行度 0、終局を進行度 128 として 128 段階に分けている。

何故進行度を考慮しなければならないのか。将棋において、序盤はあまり局面の良し悪しがはっきり出ることは少ない。序盤は定跡と呼ばれる昔から研究されてきた『手順どおりに打てば不利にならない手』というものが存在するからである。またたとえ不利になる手を打ったとしても中盤以降の打ち方によっては十分逆転することが可能な場合が多い。すなわち、序盤においての評価値の差というものはよほど大きな差でない限りひっくり返すことが可能であるために信用度が低いということになる。特に評価値が 0 に近い場合、評価関数の違いによって有利不利の判断が変わってしまうことも少なくないため十分に信用することが

できないことが多い。また、序盤では打つことができる手が非常に多いために探索木が大きくなってしまうことも分かっている。そのため時間内に十分な深さまで探索することができず、後々に不利になってしまう手を打つ可能性も大きいのである。

局面が中盤に入ってくると評価値の値は信用できるものになってくる。中盤では序盤と異なり駒の取り合いや攻め合いといった評価値の大きく変化する手が指されるため、盤面の形によってある程度はっきりと有利不利をつけることができるからである。中盤になると序盤に比べて探索すべき手が絞られてくるため序盤ほど探索木は大きくならないことも評価値が信用できるものになってくる理由の一つである。

終盤になると評価値はほぼ間違いのないものになる。コンピュータのプレイヤは手が限定されてくるほど探索木が小さくなって探索量が減るため、正確に最善手を見つけられるようになるからである。特に詰みを探すような局面においては人間のプレイヤを上回っていると評価されている。よって、終盤の評価値はそのまま勝敗に結びつくものとして扱えると考えられる。

3.2.3.2 実験

3.2.3.1 で述べた進行度との依存性をふまえて評価値と勝率の関係を実験により求めた。実験の条件は以下の通りである。

- サンプル局面の進行度
  - 序盤 32
  - 中盤 64
  - 終盤 96
- 探索の深さ
  - 16
- 実験方法
  - 既存の棋譜を読み込ませ指定の進行度での局面を探索し評価値を求めた
  - 評価値を 100 毎の区間に分割し範囲内の局面数をカウントした
  - 探索を行った対局で最終的に勝った対局数をカウントした

対局が始まった直後は定跡とのパターンマッチングを用いるためコンピュータは探索を行っていない。そこで、定跡を外れた直後に近い進行度 32 での局面を序盤のサンプルとした。中

盤は 128 段階の中央である進行度 64 の局面をサンプルとした。終盤は、詰みの探索に入ってしまうと通常の探索と異なる探索アルゴリズムを用いてしまうので、詰みを見つける直前に近い進行度 96 での局面をサンプルとした。探索の深さについては、ふかくまで探索しすぎると実験に大変時間がかかってしまうことを考慮して深さ 16 とした。深さ 16 であれば通常の対局に近い深さまで探索を行っているので十分な探索を行ったと言える。実験の結果を以下に示す。

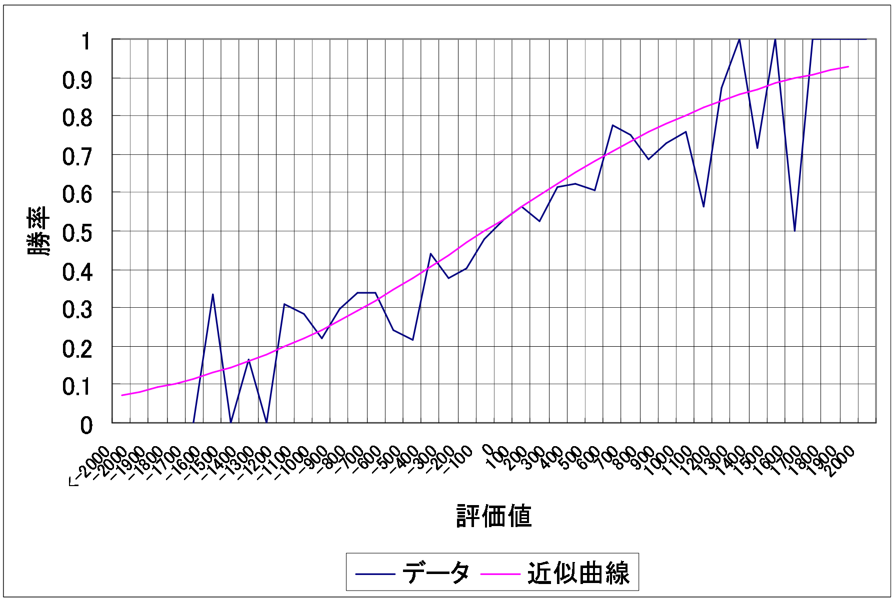

図 3.6: 評価値と勝率の関係（進行度 32）

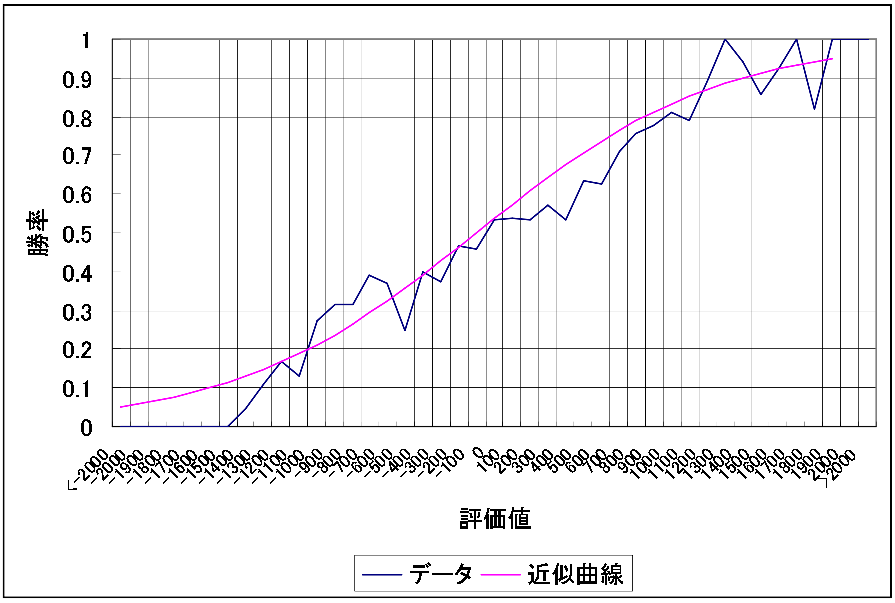

図 3.7: 評価値と勝率の関係（進行度 64）

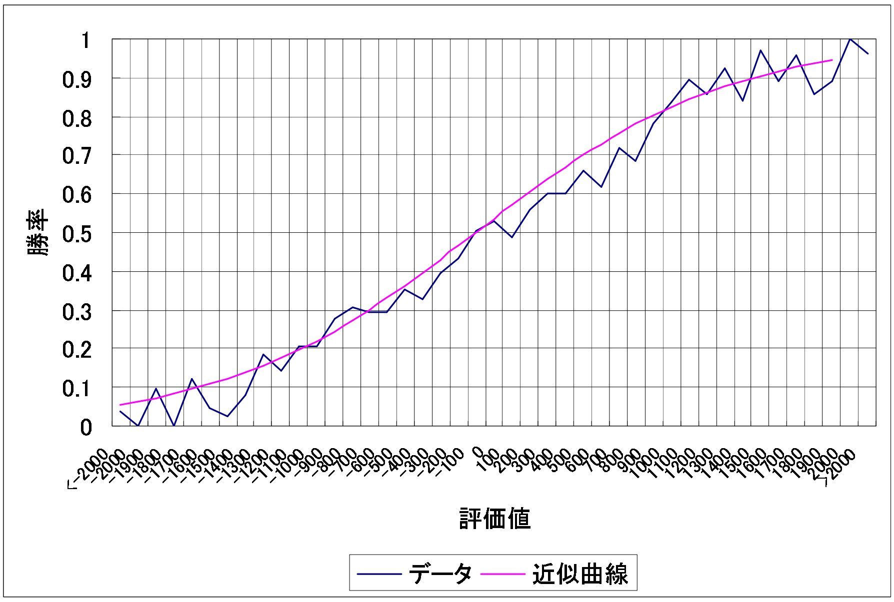

図 3.8: 評価値と勝率の関係（進行度 96）

図 3.6 から 3.8 はそれぞれの進行度における評価値と勝率の関係を表している。青い折線がデータを結んだ線で、紫の線が近似曲線である。評価値が完全に正確なものであるとすると、プラスであれば自分の勝ち、マイナスであれば自分の負けとなるはずなので、評価値と勝率の関係のグラフはステップ関数を用いて表現できるはずである。しかし、実際の評価関数には必ずエラーが含まれているためにステップ関数にはならず、それに近い形になると考えられる。そこで本実験においては結果をシグモイド関数で近似することにした。シグモイド関数とは

$$f_{(x)} = \frac{1}{1 + \exp^{-x}} \tag{3.2}$$

で表される関数である。無限回微分可能であるという便利な性質を持つため、多くの手法で取り入れられている。これはコンピュータゲームプレイヤの世界においてもよく使われている。本研究でもこのシグモイド関数を用いることで評価値と勝率の関係を近似することにした。進行度による評価値と勝率の関係式の変化は次のようになった。

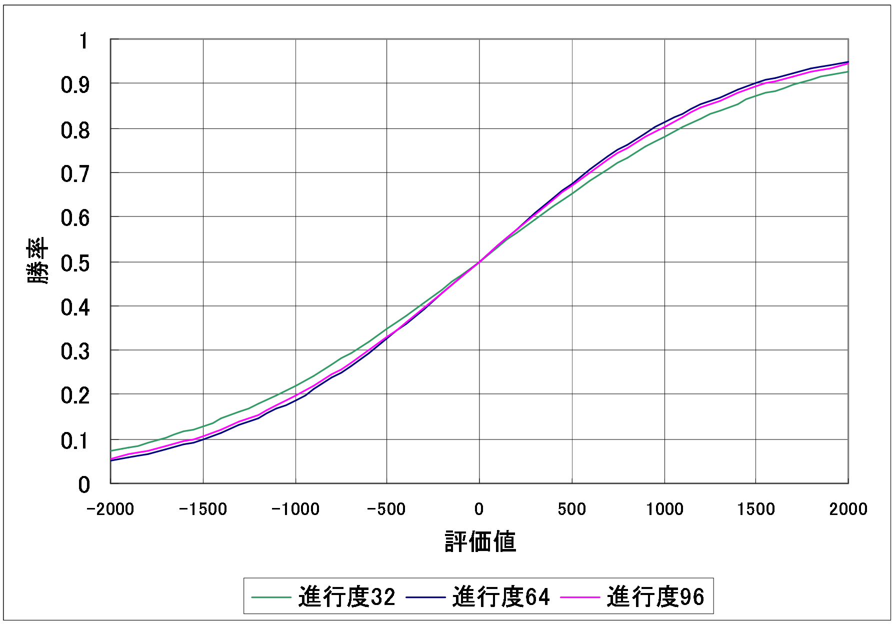

図 3.9: 進行度による関係式の変化

グラフを見ると、進行度 32 ではシグモイド曲線が若干なだらかになっていることが分かる。進行度 64 と進行度 96 はほぼ同じ曲線となった。予想では進行度 96 の方が進行度 64 の式よりステップ関数に近くなると思われたが、実際には進行度 64 の方が近くなる結果となった。これは、局面の評価値が非常に高いにもかかわらず相手に詰みを見つけられてしまい逆転負けをするケースがいくつか存在したのが原因だと考えられる。同様に、評価値が非常に

進行度	x の係数
32	-0.001269
64	-0.001461
96	-0.001410

表 3.2: 進行度と勝率関数の係数

低い時にこちらが詰みを見つけて逆転するケースも存在したため、近似曲線は進行度 64 のものよりもなだらかになってしまったのだと考えられる。

ここで求めた近似式は、4 で激指に実装した手法の中で、現在探索している局面での勝率を計算するのに用いた。

第4章　実装

本章では 3 で述べた勝率を用いた探索時間の最適化手法の激指への実装について述べる。4.1 では、提案手法をどのように激指へ実装したかを説明する。4.2 では、提案手法を実装した激指とオリジナルの激指との対局を行いどれだけ探索時間を最適化できているか、また強さがどれだけ変化したかを述べる。4.3 では、4.2 で示した実験結果をふまえて考察を述べる。

4.1 実装方法

3 で述べた勝率を用いた探索時間の最適化手法を実装した。3.1 で述べたように、探索時間の最適化にはマクロな視点からの最適化とミクロな視点からの最適化の二つの層が存在する。ここでは、そのそれぞれについてどのように実装を行ったのかを説明する。

マクロな視点からの最適化 マクロな視点からの探索時間の最適化は、一局の対局を通した中のどの局面で長く探索を行えば良いかを求めるものであることを 3.1.2.1 で述べた。また、対局が終了してからであれば結果的にどの局面で長く探索を行うべきだったかを求めるのは比較的容易であると考えられるが、対局をしている最中に長く探索を行うべき局面であることを判断するのは非常に難しいことも述べた。本実験では、探索の反復深化中に手が頻繁に変わるようであれば評価値の前後する手で迷っているものだと考えられるため、探索時間を長くしてどちらかの手に絞るという手法をとることにした。実装の詳細は以下のようになる。

- 探索の反復深化中に各深さでの最善手を記録しておく
- 最善手が連続である回数以上変化した場合探索時間を延長する
- 延長する前の探索時間はオリジナルのプレイヤと同じ長さとする

探索時間を延長する手の変化回数はいくつか値を試してみることにした。探索時間を延長する場合はオリジナルの激指で基準となっている 30 秒を一つの単位として 30 秒ずつ探索時間を延長していった。

ミクロな視点からの最適化 ミクロな視点からの探索時間の最適化は、その局面でどれだけ長く探索を行えば良いかを求めるものであることを 3.1.2.2 で述べた。しかし、3.2.1 において探索時間を長くしても評価値および勝率が上がるとは限らないことが分かったため、探索時間を長くすることによる勝率の上昇を用いて勝率を最大化する計算を行うことは難しい。そこで、ある程度探索時間を延長してから探索時間の延長前と延長後の勝率、および延長前と延長後の最善手を比較することによって、そこで探索を止めるかどうかを判断することにした。実装の詳細は以下のようになる。

- 延長後の最善手が延長前と同じ場合
  - 手が安定したと考え探索を終了する
- 延長後の最善手が延長前と異なる場合
  - 延長前と延長後の評価値による勝率を計算
  - 求めた勝率を比較する

    勝率が上がった場合
      手が変わっておりかつ探索の余裕があると判断し探索を続ける

    勝率が下がった場合
      これ以上の探索を行うと残り時間が減るだけと判断し探索を終了する

局面の勝率を求めるのには 3.2 で求めた探索時間と勝率の関係および評価値と勝率の関係を用いた。3.2.3 で求めたように、勝率はシグモイド関数で近似している。係数は現在の局面の進行度を元に表 3.2 に示した値を用いた。実験では、定跡を外れてから進行度 63 までを序盤、進行度 64 から進行度 95 までを中盤、進行度 96 以降を終盤として扱った。勝率の計算式は以下のようになる。

$$W = \frac{1}{1 + exp^{phase*val}} * p_{win} \tag{4.1}$$

W を勝率、$phase$ を進行度別の係数、val を評価値、p_{win} を残り時間による勝率の補正値とする。勝率の補正値とは、対局相手と自分の残り時間の差による勝率の変化を表す。長く考えると残り時間が少なくなり不利になることは 3.1 の中で既に述べた。よって残り時間が少ないほどプレイヤは弱くなるので、残り時間の量を考慮して勝率に補正をかけなければならないことは明らかである。そこで、現在の残り時間を予想される残り手数で割ることによって残りの局面で一手にかけられる探索時間を概算し、その一手当たりの探索時間の違いによる勝率の補正をかけることを行った。

4.2 実験とその結果

実験では提案手法を取り入れた激指とオリジナルの激指で対局を行いその勝率を計算した。探索する深さの初期値を 14 から 16 まで変化させ、対局中のログから探索時間の変化を調べた。オリジナルの激指は最大探索時間が 30 秒となっているのに対し、提案手法を取り入れた激指は探索時間の上限は定めず、4.1 で述べた方法により探索時間を決定した。

実験の結果を以下に示す。

対局開始時の深さ	勝敗数	提案手法を実装したプレイヤの勝率
16	38 勝 45 敗 1 分	0.4523
15	51 勝 52 敗 0 分	0.4951
14	79 勝 72 敗 2 分	0.5163

表 4.1: 提案手法を実装したプレイヤの勝率

提案手法を実装したプレイヤの勝率は表 4.1 のようになった。これをみると、探索の深さが元々深い場合には今回の提案手法が悪影響を及ぼしてしまっていることが分かる。それに対し、探索の深さが元々浅い場合にはあまり影響を与えていないことが分かる。これについてもう少し詳しく見てみよう。以下に、各深さでの対局の例のグラフを取り上げる。

図 4.1 を見ると、30 秒以上探索を行っている手が特に中盤以降に多く見受けられる。しかし、60 秒以上探索を行っている手は一つも無かった。図 4.2 を見ると、序盤と終盤に 30 秒以上の探索を行っている手はほとんどなく、中盤に長い探索が固まっている。図 4.3 では全体的に長い探索を行っている手が少ない。中盤以降にいくつか長い探索を行っている手があるが、その数は少ない。

以上の結果を見ると、探索があまり深くない状態では本研究の提案手法による探索時間の延長は適用されることが少ないと見ることができる。これが深さ 14 で対局を始めた時の方が深さ 16 で始めた時と比べて勝率に影響が出ていない原因だと考えられる。深さ 16 で勝率が若干下がってしまった原因は、的確な位置で探索時間を延長するはずが必要以上に多くの局面で長い探索を行ってしまった結果、残り時間が少なくなって弱くなってしまったものだと考えられる。

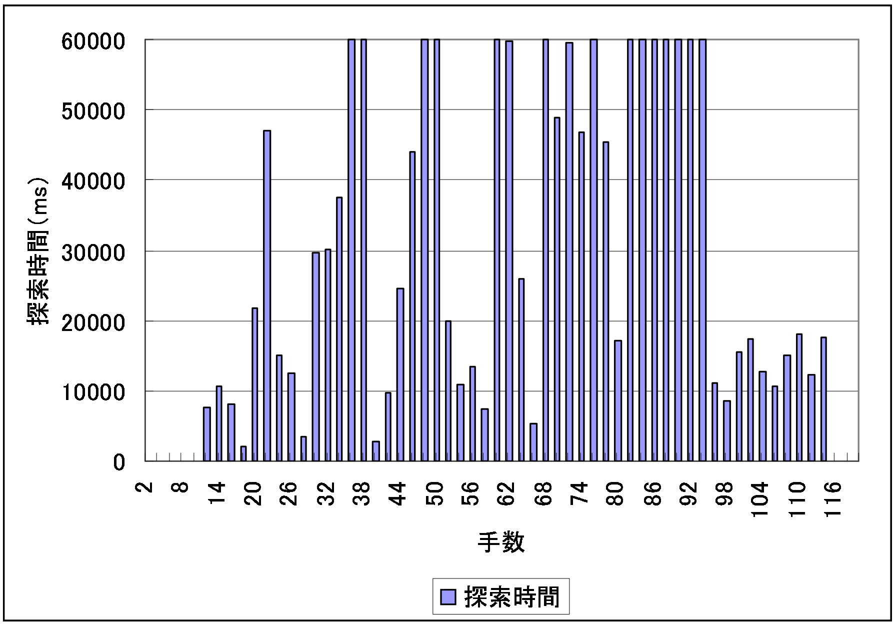

図 4.1: 深さ 16 での対局時の探索時間

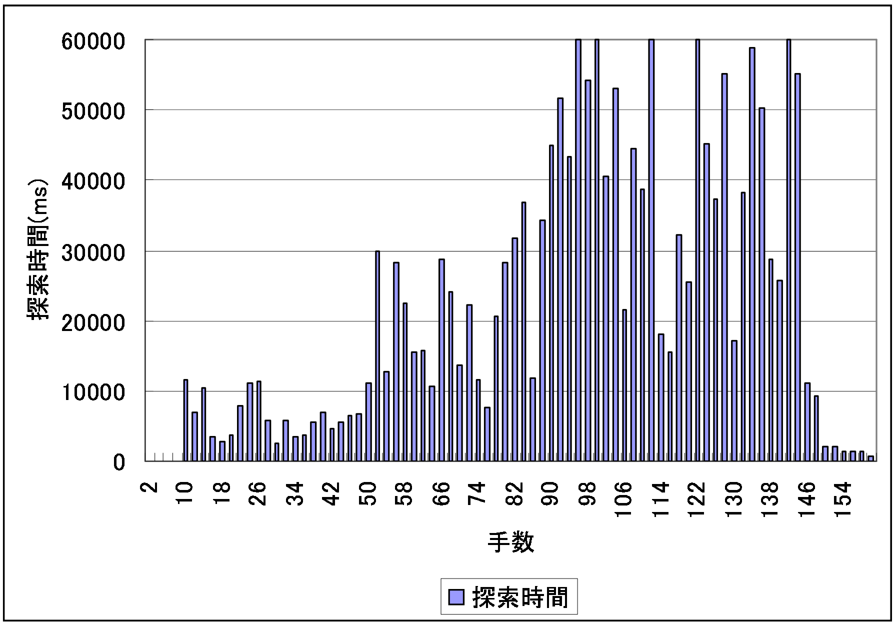

図 4.2: 深さ 15 での対局時の探索時間

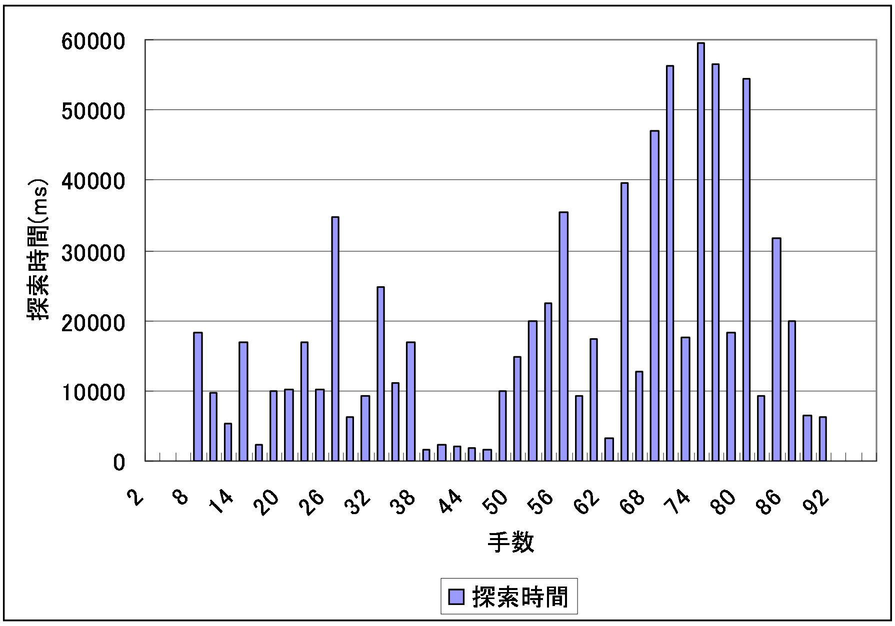

図 4.3: 深さ 14 での対局時の探索時間

4.3 考察

対局数が少ないため統計的なデータとしての信頼性は高くはないが、今回の実装方法ではオリジナルとほとんど強さが変わらないかもしくは弱くなってしまう結果となった。この実験結果から、今回提案した手法のままではコンピュータゲームプレイヤを強くすることには貢献できないことが分かった。本実験から考えられる問題点は次の四つである。

- 持ち時間切れ負けというコンピュータ将棋のルールの特殊性を考慮していない
- 探索時間を延長する判断基準に問題がある
- 最大で探索時間が 60 秒までしか延長されていない
- 探索時間をどれだけ延長するかを正確に計算できていない

以下に、これらの問題点についての考察を述べる。

持ち時間切れ負けというコンピュータ将棋のルールの特殊性を考慮していない 人間の対局では、持ち時間が無くなった場合は『秒読み』といって決められた秒数以内に手を打たなくてはいけない状態になる。しかし、自分の持ち時間がなくなっても負けにはならない。これに対しコンピュータ将棋で最も多く用いられているルールでは、自分の持ち時間が無くなった時点で『切れ負け』として負けになってしまう。そのため、人間のプレイヤが持ち時間を全て使って考えることができるのに対し、コンピュータのプレイヤは必ず残りの持ち時間を残しておかなければならない。また、どんなに早く一手を打ったとしても必ず 1 秒にカウントされてしまうため、残りの手数が分からない以上、切れ負けにならない十分な時間を残して探索を行わなければならない。そのため激指では残り時間が少なくなってくると探索の深さを減らすことによって早く探索を終わらせるように調整されている。そのため、探索の深さが浅くなってくると探索時間の上限を上げても時間内に探索が終わってしまい、探索時間の調整が無効となってしまう。よって中盤までは長く考えるが終盤以降は長く考えずに探索が終わってしまうという状態になってしまい、探索時間を長くした効果が出なかったと考えられる。

探索時間を延長する判断基準に問題がある 3.1 で述べた方法は、評価値が大きく動いている局面においては効果があると考えられる。評価値が大きく動いて最善手が変化している場合、評価値が前後しない安定した局面まで探索を行うことは効果的である。しかし今回は最善手の変化のみで判断を行っていたため、評価値が大きく動いているかどうかの確認ができなかった。この場合、安定した局面ではあるが非常に評価値が近いために最善手が入れ替わってしまっているケースでも探索時間を長くすることになってしまう。安定した局面で評価値が近い場合はどちらの手を選んでも大差の無いことが多いため、結果的に探索時間を無駄に使ってしまっていることになる。その結果、他の局面で十分な探索が行えなくなってしまったケースがあると考えられる。

これを改善するためには、最善手の変化だけでなく探索している手の局面の評価値の変化が大きいかどうかを調べなくてはならない。また、本実験では最善手が 4 回連続で変わった時に探索時間を延長することにしたが、この 4 回という数字も過去の激指同士の対局の棋譜から『3 回以下の手の変化は頻繁に起こりうるので除外する』と経験的に導き出した物であり、具体的な数値を元に求めたものではないために信頼性が薄いといえる。

最大で探索時間が 60 秒までしか延長されていない これは先に述べた探索時間を延長する判断基準に問題があることに関係してくると考えられる。本実験での探索時間の延長を続ける基準は

- 探索時間を延長した後の最善手が延長する前と異なる
- 探索時間を延長した後の方が延長する前よりも勝率が高い

という条件であった。しかし探索時間を長くした場合 3.1.2.2 で述べたように残り時間が少なくなった分だけ勝率が下がることになる。本実験の場合、探索時間を長くすることによる勝率の低下が大きく、何度も探索時間を延長することなく探索を打ち切ってしまったのだと考えられる。また、激指は持ち時間が残り少なくなってくると探索の深さを浅くして時間調整を行っている。そのため持ち時間が少ない状況では探索が早めに終了してしまい、探索時間をそれ以上延長することなく最善手を返してしまうことがあった。これは先にも述べた『切れ負け』というルールの関係上、避けられない問題であると考えられる。よって、残り時間が十分あるときに一手毎にどれだけ有効な時間配分ができるかどうかに焦点を当てるべきだと考えている。

探索時間をどれだけ延長するかを正確に計算できていない 本実験では 30 秒を一つの単位として探索時間を調節していった。しかし本来は 30 秒といった固まりではなくもっと連続的に探索時間を最適化していくべきである。この問題は、本実験のようにまとめて探索時間を延長するのではなく、探索時間を延長した時の勝率の上昇値や局面の特徴量などから最適な探索時間を具体的に計算することで解決できると考えられる。これらの情報は求めることが難しい。長く探索を行うべき局面の特徴量は持ち時間の使い方の詳細が記録されている棋譜が存在すれば学習することは可能であると考えられるが、持ち時間の詳細が記録されている棋譜を大量に用意することは難しく、多くの時間がかかってしまう。また、探索時間を延長した時の勝率の上昇値はについては、探索を長くすることが勝率の上昇に直接関係していないため、計算によって具体的な勝率の上昇値を求めることは難しい。これらの課題をクリアすることができれば、もっと正確に一手にかけるべき最適な探索時間を計算することができるようになると考えている。

第5章　まとめと今後の課題

5.1 まとめ

本研究では探索時間の適切な割り当てができていないコンピュータゲームプレイヤに対し、勝率という概念を用いることで最適化された探索時間を割り当てようという試みを行った。また、探索時間と勝率の関係や評価値と勝率の関係を実験から求め、得られた結果を元に探索時間の最適を行った。

最適化のために必要なデータとして、、2.2.2 で説明した実現確率打ち切り探索の特徴である連続的に可変な探索の深さを利用するために 2.3.2 にて探索時間と実現確率の閾値の関係を求めた。また、3.2.1 において探索時間と評価値の関係を、3.2.2 において探索時間と勝率の関係を、3.2.3 において評価値と勝率の関係を求めた。　残念ながら 4.2 で述べたとおり、結果としてオリジナルのプレイヤと変わらないか、若干弱くなるという結果に終わってしまった。本研究で用いたコンピュータ将棋プレイヤ『激指』は探索の深さや探索時間の上限等が非常に高い精度で的確に設定されているために、本研究の提案手法による探索時間の決定方法ではより強いプレイヤにすることができなかった。

本研究での実験における知見としては次の四つが挙げられる。

- 探索時間を延長した場合と延長しなかった場合の局面の評価値変化データの必要性
- 探索時間を延長した場合の具体的な勝率の上昇値の必要性
- 進行度別における探索時間延長回数データの必要性
- 探索の深さの違いによる探索時間延長による探索への影響の程度

今回の実験では探索時間を延長した場合と延長しなかった場合の最善手の変化を追って探索を継続するかどうか判断していた。しかし、実際には探索した局面自体の状態も探索を延長するかどうかに大きく関わってくることが分かった。つまり今回の手法は評価値変化の激しい局面で最善手を迷っている場合においては効果的だが、拮抗した局面で最善手を迷っている場合にはあまり効果がないということである。次に、探索時間を延長した場合の具体的な勝率の上昇値の必要性だが、3.2.1 において探索時間が長くなるほど評価値が安定することを述べた。しかし、これはあくまで局面の評価が安定するだけであって、具体的にどれだけ勝率が上昇するのか求められていない。このデータは探索時間を最適化する際に必要なデータであり、一手当たりにどれだけ探索時間を割いて良いか計算するために必要となってくる。進行度別における探索時間延長回数データについては、進行度の違いによって探索時間を長くする回数が違ってくることから、進行度と長く探索を行うべき局面に何かしらの因果関係が存在すると考えられる。このデータを取ることで、現在の局面が長く探索すべき局面かどうかを決定する精度が上がると考えられる。最後に、探索の深さの違いによる探索時間延長による探索への好影響の程度だが、4.2 で見られるように、元の探索の深さが深ければ深いほど探索時間を伸ばしたときの影響が強い。逆に、探索時間が元々短い探索の深さで

あれば定められた時間内に探索しきることができるため、探索時間を延長しても効果が薄いことが分かった。

5.2 今後の課題

今後の課題としては以下の様なことが挙げられる。

探索時間を長くするべき局面の特徴量の抽出 4.3 で述べたように、反復深化中の最善手の変化だけで探索時間を延長すべきかどうかを判断するのは困難であることが分かっている。そこで、今回は得ることができなかったが、一手にどれだけ時間をかけたか詳細に記録してある棋譜を準備し、長く考えている局面における様々な特徴量を取るべきであると考えている。

探索時間を長くした時の勝率の上昇値の導出 探索時間を長くしても勝率が上がることと直接関係がないことは何度も述べた。しかし、長く探索すべきと判断された局面において最適な探索時間を決定するためには探索時間を長くした時にどれだけ勝率が上昇するかが分かっていなければならない。そこで、この勝率の上昇値を導出するための方法を考えることが必要である。その方法の候補としては、探索時間の延長前と延長後の同じ手の評価値の変化を調べるという方法や最善手とその他の手の評価値の差がどれだけ広がるかを調べる方法がある。

より軽い探索時間と深さの関数の作成 本研究では探索時間と実現確率打ち切りの深さの関係は 2.3.2 で求めた関係式を用いている。この関係式は信頼性は高いが煩雑な値を取っているため、探索中で計算するときには高コストの計算になってしまっていると考えられる。そこで、探索に影響を与えない低コストな計算で済む関係式を作成したいと考えている。

参考文献

[1] Matt Ginsberg. *Essentials of Artificial Intelligence*. Morgan Kaufmann Publishers, 1993.

[2] Andreas Junghanns. Are there practical alternatives to alpha-beta in computer chess? *ICCA Journal*, Vol. 21, pp. 14–32, 1997.

[3] 鶴岡慶雅 横山大作 丸山孝志近山隆. 局面の実現確率に基づくゲーム木探索アルゴリズム. *Game Programming Workshop 2001*, Vol. 14, pp. 17–24, 2001.

[4] 金沢伸一郎. 金沢将棋のアルゴリズム. 松原仁編著（編）, コンピュータ将棋の進歩 3, pp. 15–26. 共立出版, 5 2000.

[5] 山下宏. 0.5 手延長アルゴリズム. *Game Programming Workshop '97*, pp. 46–54, 1997.

[6] 橋本剛. 実現確率探索のゲーム全般への応用-line of action を題材にして-. *Game Programming Workshop 2002*, pp. 81–86, 2002.

発表文献

[1] 阿部崇史, 横山大作, 近山隆. 勝率を用いた探索時間最適化手法. 第 11 回ゲーム・プログラミング ワークショップ, pp. 159–162, 2006.

謝辞

本研究を進めるにあたり、多くの方にお世話になりました。

近山隆教授、田浦健次朗助教授には、論文作成やプレゼンテーションの方法など多くのご指導、ご教示を頂きました。横山大作助手には、研究生活から研究内容、その進め方まで多岐にわたって日頃から相談に乗って頂きました。他の近山・田浦研究室の皆様にも公私にわたり多くの助言を頂きました。

ここに、心より感謝の意を表します。

平成 19 年 1 月 31 日