モンテカルロ木探索によるコンピュータ将棋

佐藤 佳州[†1] 高橋 大介[†1]

本論文では，モンテカルロ木探索によるコンピュータ将棋を実現し，その有効性を検証する．モンテカルロ木探索によるゲームの実現は，ゲームプログラミングの分野において現在最も注目を集めているテーマの一つであるが，将棋では今のところよい結果を得ることには成功していない．本研究では，コンピュータ囲碁で成功した手法を元に，キラームーブの導入など将棋向けの改良を加えたモンテカルロ木探索によるコンピュータ将棋を実装した．

次の一手問題による性能評価では，アマチュア初段程度のプログラムに迫る正答数を得ることに成功し，モンテカルロ木探索が将棋においても有効であることを示した．現在のトップレベルの将棋プログラムはプロに迫るまでとなっており，モンテカルロ木探索のみにより従来からの手法を単純に上回る棋力を得ることは難しいと考えられる．しかし，序盤の定跡選択や一部終盤では従来の手法よりも良い結果を得ることに成功し，モンテカルロ木探索がコンピュータ将棋においても大いに有望であることを示した．

A Shogi Program Based on Monte-Carlo Tree Search

Yoshikuni Sato[†1] and Daisuke Takahashi[†1]

Recently, Monte-Carlo Tree Search is attracting much attention in game programming. This method has succeeded in Computer-Go, however it has not yet been able to attain good result in Computer-Shogi. We implemented a Shogi program based on Monte-Carlo Tree Search, using techniques proved in Computer-Go with improvements for Computer-Shogi.

In the results of solving problems, the number of correct answers that our program found was almost the same as that of about amature 1-dan program. Although the strength of top-level Shogi programs are almost as strong as professional players, our method based on Monte-Carlo Tree Search achieved better performance than existing methods in openings and some positions of end-game. The results of our experiments showed that Monte-Carlo Tree Search holds great promise in Computer-Shogi.

1. はじめに

将棋，チェス，オセロといった思考ゲームをコンピュータで実現する際には，探索と評価関数を用いるのが一般的である．一方，現在注目を集めている手法として，モンテカルロ木探索（Monte-Carlo Tree Search）[1)2)] がある．この手法の最大の特徴は，評価関数が不要となることであり，その設計が困難とされている囲碁では大きな成功を収めた．

コンピュータ将棋は，手作業や学習による評価関数がある程度成功を収めており，アマトップクラスの性能を持つまでになっている．しかし，将棋においても万能な評価関数の設計は難しく，複雑な終盤や不利な局面では最善手が選択できないことが多い．また，序盤の指し手選択も，定跡データベースとの一致のみを見ている場合が多く，課題の一つと言える．

モンテカルロ木探索は，これらの問題点を解決できる可能性のある手法として将棋の分野でも注目を集めているものの，今のところよい結果を得ることには成功していない．モンテカルロ木探索によりコンピュータ将棋を実現する際の特有の問題としては，ランダムな指し手の選択によるシミュレーションでは終局条件（詰み）を満たすことが難しい，ゲームの性質として囲碁よりも読みに重点が置かれる，といった点があげられる．

本論文では，モンテカルロ木探索のアルゴリズムとして UCT（Upper Confidence bounds applied to Trees）[3)] を採用したコンピュータ将棋を実装する．さらに，最近のモンテカルロ囲碁で成功した手法を元に，いくつかの将棋向けの改良を行うことで前述した問題点の改善を試みる．また，定跡選択部や探索と評価関数による手法が苦手とする局面におけるモンテカルロ木探索の有効性についての検証を行う．

†1 筑波大学大学院システム情報工学研究科
Graduate School of Systems and Information Engineering, University of Tsukuba

2. 関 連 研 究

モンテカルロ法によるゲームの実現は，コンピュータ囲碁では比較的古くから研究されていた手法である[4]．この手法では乱数を利用したゲームのシミュレーション（以下 playout）を何度も行い，そのスコア（石の数の差，勝率など）によって局面を評価する．モンテカルロ法によるゲームでは，人間の知識に基づいた評価関数を設計する必要がないため，その設計が困難とされていた囲碁において大きな注目を集めた．しかし，相手のミスを期待した手を選択してしまう，ある程度以上 playout を増やしても棋力が頭打ちになってしまう，といった問題があり従来の手法を上回るには至らなかった．

モンテカルロ法に木探索を組み合わせることによりこれらの問題点を大幅に改善し，コンピュータ囲碁にブレイクスルーをもたらした手法がモンテカルロ木探索である．モンテカルロ木探索の出現により，アマチュア級位者レベルであったコンピュータ囲碁は，9 路盤ではプロに勝利するまでの強さになった．モンテカルロ木探索の最も一般的なアルゴリズムとしては，UCT (Upper Confidence bounds applied to Trees)[3] があげられる．UCT は，その時点までに行った playout の回数と勝率に基づき，より有望な指し手（ノード）に多くの playout を割り当て，探索木を成長させていく．Playout 中の指し手の選択は，完全なランダムでは良い性能を得ることは難しく，多くの囲碁プログラムでは，ゲームの知識を利用して確率的に行っている[5)6)7]．

モンテカルロ法によるコンピュータ将棋の研究としては文献 8) があげられる．この研究では，囲碁プログラム CRAZY STONE[1] のアルゴリズムを将棋へ適用し，モンテカルロ法によるコンピュータ将棋の可能性を考察している．Playout に長手数を要するという問題に対しては，最大手数を 10 程度に制限し，末端で評価関数を利用することにより解決を試みている．しかし，次の一手問題の正答率は 3%程度にとどまっており，有望とは言いがたい結果となっている．

本論文では，探索手法として UCT を採用する．さらに，キラームーブやヒストリーヒューリスティックの利用，囲碁において成功している progressive widening[5] を将棋向けに適応，並列化などを行うことにより性能の改善を試みる．また，playout 部分では最大手数を極端に制限することや評価関数を利用するといったことは行わず，ヒューリスティックの導入により自然な終局の実現を目指す．

3. モンテカルロ木探索によるコンピュータ将棋

モンテカルロ木探索は playout 部分と探索部分に大きく分けられる．本章では，それぞれについての改良点を述べる．また，定跡選択に特化した実装についても説明する．

3.1 Playout の改良

Playout 中の指し手の選択は完全にランダムでは良い性能を得ることは難しく，乱数を利用しつつも，何らかの形でヒューリスティックを導入し，対象とするゲーム（囲碁，将棋）らしい指し手の選択を行う必要があることが経験的に知られている．また，終局までに長手数を要するといった将棋特有の問題点もヒューリスティックの利用により改善することが期待できる．

本論文では指し手の選択に，Elo レーティングを利用した手法[5)9] を用いた．この手法は，指し手をいくつかの特徴の集合とみなし，実戦譜を元にその選択されやすさを Elo レーティングとして数値化し，その値に基づいて指し手の選択確率を決定するというものである．

3.1.1 Bradley-Trerry モデルと Elo レーティング

Bradley-Terry モデルはある試合に関する勝敗を予測するモデルである．あるプレイヤ i が γ_i という正の値（レーティング）をもつとすると，1 から n 人までのプレイヤの中で $i\,(1 \le i \le n)$ が勝利する確率は，

$$P(i\,\text{が勝つ}) = \frac{\gamma_i}{\sum_{j=1}^{n}\gamma_j}$$

と表わされる．複数プレイヤによるチームでの試合は，チームの強さをチームを組むメンバの強さ γ_i の積とすることで，扱うことができる．プレイヤ $1, 2, 3, 4, 5, 6$ が存在し，1-2-3, 4-5, 6 というチームを組むとき，チーム 4-5 が試合に勝つ確率は，

$$P(\text{チーム 4-5 が勝つ}) = \frac{\gamma_4\gamma_5}{\gamma_1\gamma_2\gamma_3 + \gamma_4\gamma_5 + \gamma_6}$$

とモデル化される．一般的には，γ_i を $400\log_{10}\gamma_i$ と変換したものを Elo レーティングと呼ぶが，本論文中では，変換前の値 γ_i をレーティングの値として扱う．

3.1.2 将棋への適用

将棋の指し手には,「駒の損得」,「王手」や「逃げる手」など様々な特徴がある．これらの特徴を個々のプレイヤとみなし，指し手を特徴からなるチームと考えることで前節のモデルを適用できる．用いた主な特徴を以下に示す.

- 駒の損得（SEE）
- 駒をとる手
- 成る手
- 王手
- 相手の利きのある駒の移動（逃げる手）
- 位置テーブルの値の増減（移動時，打つ場合）
- 玉の位置，周囲の利き

- n 手前に移動した駒との関係

駒の損得はSEE（Static Exchange Evaluation）により求めた．駒の価値は**表 1** に示す値を用いた．

表 1　駒の交換値

歩	香	桂	銀	金	角	飛
100	280	300	420	530	620	700
と	成香	成桂	成銀	－	馬	龍
270	320	250	430	－	710	850

駒の損得（SEE）の計算は，他の特徴に比べコストがかかるが，将棋において特に重要な特徴と考えられる．

位置テーブルの値は文献 10) の手法により求めたものを用いた．**図 1** は玉が８八にいる場合の味方の金の位置による点数である．

-62	-108	-64	-70	-88	-98	-88	-124	-98
-116	-112	-4	4	-22	-28	-6	-74	-94
-110	-46	-50	-26	-22	-2	-28	-44	-76
-64	-40	0	-16	10	-18	-8	-4	-50
-98	-52	-26	-18	-22	-14	-20	-26	-60
-78	-38	-20	-26	-20	-18	-14	-44	-66
-104	-20	2	18	-48	-22	-52	-64	-100
-120	X	46	-22	-2	-34	-52	-80	-162
-116	6	-58	8	-82	-30	-86	-74	-160

図 1　玉が８八にいる場合の金の位置による点数

矢倉囲いや美濃囲いの金の位置の点数が高くなっており，玉から遠くなるほど減点されていることが分かる．このような表を自玉，相手玉がそれぞれ１一から９九にある場合について作成している．

3.2 UCT の改良

UCT では通常以下の式（UCB1[11]）の値を最大とする子ノードを選択する．

$$X_i + c\sqrt{\frac{2\log n}{n_i}} \tag{1}$$

X_i はノード（指し手）i を選択した場合の勝率，n, n_i はそれぞれ親ノード，ノード i の訪問回数を示す．c の値は通常は 1.0 を用いる．1.0 より大きくした場合には勝率の低い手に，小さくした場合には勝率の高い手により多くのシミュレーションが割り当てられることになる．

UCT の大きな問題点は，訪問回数の少ないノードにおいて効率のよい割り当てができず，ランダムに近い選択になってしまうことである．本論文では，キラームーブやヒストリーヒューリスティックの利用，progressive widening などによりこの問題を改善した．

3.2.1 キラームーブ，ヒストリーヒューリスティックの利用

本論文では，UCT の子ノードの選択にキラームーブ及びヒストリーヒューリスティックを利用した．ともにコンピュータチェスやコンピュータ将棋においてよく用いられている手法で，キラームーブは兄弟ノードの最善手，ヒストリーヒューリスティックはある指し手が最善手であった割合（回数）を示す．将棋では，ゲームの性質上多くの兄弟ノードでは最善手が同一であることが多く，キラームーブは特に重要な概念であるといえる．

キラームーブやヒストリーヒューリスティックの値が大きいノードでは，式 (1) における c の値を 1.0 よりも大きく設定し，訪問回数が少ないうちは他のノードより優先的に選択されるようにしている．

3.2.2 Progressive widening

Progressive widening[5] とは，新しいノードを作成する際，一度にすべての合法手を探索対象とするのではなく，ヒューリスティックを利用してよさそうな指し手から探索対象を広げていくという，一種の枝狩り手法である．

本論文では，未訪問のノードでは式 (1) における勝率の値 X_i の部分に，指し手のレーティングの値 R_i を利用した以下の式を用いることで progressive widening を実現した．

$$dR_i + c'\sqrt{\frac{2\log n}{e}}$$

R_i の値は以下の式により指し手 i のレーティングを 0 から 1.0 の値に正規化したものである．定数 c', d, e の値は実験的に 0.5, 4.0, 10.0 とした．

$$R_i = \frac{\text{指し手 } i \text{ のレーティング}}{\text{合法手中の最大のレーティング}}$$

このような式を用いることで，レーティングの高い指し手から順に探索が行われる．単純に駒得する手などレーティングの値が突出して高い手がある場合には，その手が特に重視されることになる．

本論文で用いた progressive widening は，囲碁の例と比べて弱い枝刈りとなっている．将棋の場合，一見駒損をするような手でも読みを入れることで，実は好手であることが分かるといった状況があまりに多いためである．

3.2.3 並列化，その他

モンテカルロ木探索では，playout の回数は棋力に大きく影響を及ぼす．playout の部分は独立性が高いことからも，並列化は非常に有効であると考えられる．

本論文では，探索木の部分を共有した状態で，スレッドを用い playout を独立に実行する方法を採用した．この方法では，探索木を更新するタイミングの関係上，単一スレッドで実行した場合とは異なる振る舞いをす

表 2 指し手の特徴の Elo レーティング

特徴	詳細	γ_i			
駒の損得 (SEE)	飛程度の得	20.04	位置テーブルの値の増減	歩	1.04 – 2.19
	角程度の得	14.37		香	0.16 – 0.90
	金程度の得	9.54		桂	0.57 – 2.25
	銀程度の得	6.08		銀	0.60 – 3.00
	桂香程度の得	3.89		金	0.33 – 2.22
	歩程度の得	2.55		角	0.66 – 1.45
	損得なし	1.11		飛	0.38 – 1.00
	歩程度の損	0.47		馬	0.78 – 1.49
	桂香程度の損	0.32		龍	0.66 – 0.99
	銀程度の損	0.10		その他の成駒	1.05 – 2.21
	金程度の損	0.06	位置テーブルの値（打つ場合）	歩	0.50 – 2.21
	角程度の損	0.05		香	0.24 – 0.97
	飛程度の損	0.02		桂	0.17 – 1.66
駒を取る手	取り返し	12.75		銀	0.19 – 1.35
	その他	1.88		金	0.28 – 1.45
王手		7.55		角	0.16 – 1.17
成る手		1.47		飛	0.22 – 1.89
相手の利きのある駒の移動	歩	1.13 – 1.27	玉移動（位置，周囲の利き）	通常時	0.30 – 1.01
	香	1.90 – 3.19		王手時	0.57 – 4.21
	桂	1.70 – 2.59	直前に移動した駒の移動		0.90 – 1.55
	銀	2.20 – 2.60	直前の駒との位置関係		0.89 – 1.97
	金	4.38 – 4.78	二手前の駒との位置関係		0.94 – 1.86
	角	1.77 – 5.12	直前の位置に戻る（序盤のみ）		0.10 – 0.87
	飛	6.85 – 23.68	…		…
	馬	3.26 – 9.98			
	龍	8.53 – 14.62			
	その他の成駒	0.82 – 1.65			

ることになるが，4CPU 程度の並列化においては十分な性能向上が得られることが知られている[6)].

その他の UCT の改良点としては，探索木の内部で見つけた詰みは別扱いにするといったことも行っている．

3.3 モンテカルロ木探索を用いた定跡選択

現在多くのプログラムは，定跡選択の部分に力を注いるとは言いがたく，定跡データベース（ここでは実践譜の集合とする）との一致のみで選択しているプログラムも多い．著者の佐藤が開発したプログラム「遠見」,「棋理」では，定跡データベース中のある候補手 m が選択される確率を以下の式により決定している（以下確率による定跡選択と呼ぶ）.

$$P(m\,\text{が選択される}) = \frac{m\,\text{が指された回数}}{\text{現局面がデータベースに存在する数}}$$

この手法は，多くのプログラムで用いられていると考えられる一般的な手法と言えるが，データベースの指し手を絶対的に信頼することになるため，実際には悪手であっても選択してしまうことがある，プログラムがあまり得意でない展開も選択してしまうといった問題点も存在する．

本論文では，モンテカルロ木探索を用いた定跡選択を実装し，その有効性を検証した．以下にモンテカルロ木探索を用いた定跡選択部の手順を示す．

(1) 定跡データベースに含まれる各棋譜を 1 つの playout とした UCT の探索を行う．ただし，子ノードを生成する際にはすべての指し手を生成するのではなく，定跡データベース中に存在する指し手のみを生成する．

(2) (1) で構成した探索木の上で，通常の UCT による探索を行う．

定跡選択部分では，ある程度有効な指し手が絞られており，このような状況ではモンテカルロ木探索が特に効果を発揮できると考えられる．

4. 実　　験

4.1 指し手の Elo レーティングを利用した際の playout の性質

3.1 節で述べた指し手の特徴のレーティングを**表 2**に示す．レーティングの算出には文献 5) の手法を用いた．名人戦の棋譜約 300 局を学習用データとし，各特徴のレーティングを算出した．

各特徴のレーティングは γ_i で示した値の範囲を取りうる．γ_i の値が 1.0 よりも大きい場合には選択されやすい特徴，1.0 を下回る場合には選択されにくい特徴となる．

駒の損得や直前の駒の取り返し，王手などは特に強い特徴となっており，妥当なレーティングが算出できているといえる．位置テーブルの値の増減では，飛角

より金銀などの小駒の方が，玉との位置関係が重視されていることが分かる．

表 3 に指し手の選択法による playout の性質の違いを示す．予測率は評価用の棋譜において実際に指された指し手に割り当てた確率の平均，終局率は平手の初期局面において playout を行ったとき，256 手以内に終局した割合を示す．速度は平手の初期局面において 1 秒間に実行できる playout 回数を計測したものである（実行環境は Xeon X5355，1 コアのみ使用）．

表 3 指し手の選択法による playout の性質

指し手の選択	予測率	終局率	速度
rating なし（完全乱数）	0.037	0.193	約 3500 回/秒
rating あり	0.172	0.905	約 900 回/秒

速度は 4 分の 1 程度に落ちているものの，予測率が大きく向上していることが分かる．また，終局率の向上から，終局条件を満たすことが難しいという将棋特有の問題点を改善できていることが分かる．

4.2 次の一手問題の正答数による評価

次の一手問題の正答数による性能評価を行った．結果を**表 4** に示す．問題としては，文献 12)，13) の 98 題を用いた．実験環境は Quad-Core Xeon X5355 × 2，解答時間は 1 問 30 秒とした．

MC/UCT は単純なモンテカルロ木探索（カッコ内はスレッド数），rating は playout 及び progressive widening に Elo レーティングを利用した場合，killer はキラームーブ，ヒストリーヒューリスティックを利用した場合を示す．参考として，探索と評価関数に基づくプログラムの正答数を併記した．「遠見」，「棋理」はそれぞれアマチュア初段程度，三段程度の棋力を持つ[⋆1]．「遠見」，「棋理」は詰探索ルーチンを持っているため，詰将棋を除いた問題 86 題の正答数による比較も行った．

表 4 次の一手問題の正答数

用いた手法	正答数	（除詰将棋）
MC/UCT(1)	6 / 98	6 / 86
MC/UCT(8)	12 / 98	11 / 86
MC/UCT(8), rating	41 / 98	38 / 86
MC/UCT(8), rating, killer	49 / 98	45 / 86
遠見	60 / 98	48 / 86
棋理	71 / 98	59 / 86

正答数の向上から，各手法がモンテカルロ木探索によるコンピュータ将棋において有効な改良となっていることが分かる．

モンテカルロ木探索は，詰将棋を除いた問題では探索と評価関数による初段程度のプログラムに迫る正答数を得た．詰将棋などの問題では，正確な読みが必要となるためモンテカルロ木探索には適していないと考えられる．

4.3 「遠見」との対局による評価

探索と評価関数によるプログラム「遠見」との対局を行った．結果を**表 5** に示す．思考時間はともに 1 手 10 秒，対局数は 200 局とした．

表 5 「遠見」との対局結果

用いた手法	勝率
モンテカルロ木探索	0.04
モンテカルロ木探索+静止探索	0.32

モンテカルロ木探索は，次の一手問題の正答数では「遠見」とほぼ同等であったにもかかわらず，連続対局の結果では大きく負け越していることが分かる．

モンテカルロ木探索の問題点として小さな駒損が多いことがあげられる．特に序中盤の歩損などは，playout の勝敗には結び付きにくく，意味のない歩の突き捨てなどを指してしまうことが多い．また，不利になるほど，相手のミスによる逆転に期待した悪手を指しやすくなってしまうという傾向もある．

一方，次の一手問題のような tactical な局面では，多少の駒損などは問題にならないことが多い．このようなことから，次の一手問題の正答数による評価と対局結果による評価に大きな差が出てしまったものと考えられる．

モンテカルロ木探索+静止探索は，モンテカルロ木探索に静止探索を組み合わせ，評価値がほぼ互角の場合には駒損しない手を選択するようにしたものである．単純なモンテカルロ木探索よりも勝率を改善できていることが分かる．

しかし，依然初段程度のプログラムに達することはできておらず，モンテカルロ木探索により単純に従来の手法を上回ることは難しいと考えられる．

4.3.1 プロの棋譜との一致率

モンテカルロ木探索を用いた場合のプロの棋譜との一致率を局面の進行度別に計測した．棋譜 k 中の n 手目の指し手の進行度は以下の式で示す値としている．

$$\text{進行度}\ (k,n) = 25 \times \frac{n}{k\ \text{の終局までの手数}}$$

図 2 及び**図 3** に進行度別のプロの棋譜との一致率を示す．棋譜中の指し手が，モンテカルロ木探索の最善手と一致した割合，上位 3 手に含まれていた割合，参考として「棋理」の最善手との一致率も示している．

図 2 は思考時間を 1 手 1 秒とした場合のグラフである．モンテカルロ木探索の一致率が非常に低くなっていることが分かる．これは 1 手 1 秒程度では十分な

⋆1 世界コンピュータ将棋選手権や floodgate(http://wdoor.c.u-tokyo.ac.jp/shogi/) での成績，次の一手問題の正答数から推定.

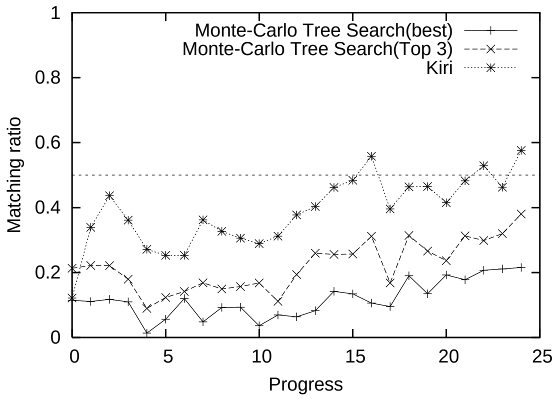

図 2 プロの棋譜との一致率（1 手 1 秒の場合）

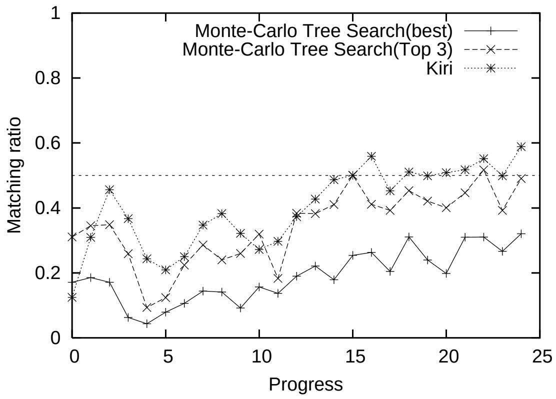

図 3 プロの棋譜との一致率（1 手 10 秒の場合）

playout を行うことができていないためだと考えられる．一方，探索と評価関数によるプログラムでは思考時間が短い場合にも，ある程度の一致率を出すことができているといえる．

図 3 は思考時間を 1 手 10 秒とした場合のグラフである．1 手 1 秒の場合に比べ，モンテカルロ木探索の一致率が大きく向上していることが分かる．一定時間内により多くの playout を行うことができれば，性能はさらに向上すると考えられる．モンテカルロ木探索の傾向として，序盤から終盤にかけて徐々に一致率が向上していく点あげられる．これは，終局（詰み）が近いほど指し手の善悪が playout の結果（勝敗）に反映されやすくなる，時間あたりに行える playout の回数が増加するといった理由が考えられる．

また，モンテカルロ木探索がプロの棋譜と一致した局面において，「棋理」がその手を選択した割合は約 0.58 となった（思考時間は 1 手 10 秒）．一致率の差から考えると低い値となっており，モンテカルロ木探索と探索と評価関数による手法は得意とする局面に違いがあると考えることができる．

4.4 モンテカルロ木探索を用いた定跡選択

モンテカルロ木探索を用いた定跡選択の評価として，確率による選択を行った場合との自己対局を行った．定跡打ち切り後の思考部には「棋理」を用いた．

結果をを**表 6** に示す．定跡選択部，通常の思考部ともに思考時間は 1 手 2 秒とした．定跡データベースとしては，プロやアマ高段者の棋譜約 3 万局のうち手番側が勝ちの棋譜のみを用いた．

表 6 確率による定跡選択との自己対局の結果

用いた手法	勝率	評価値
モンテカルロ木探索	0.61	-2.01

実験の結果，モンテカルロ木探索を用いた定跡選択が勝ち越していることが分かる．通常の探索部分や評価関数との相性の関係もあるため，モンテカルロ木探索を用いた定跡選択が直ちに優れていると言い切ることはできないが，有力な一つの手法であると考えられる．

評価値は，定跡が打ち切られた時点での局面における「棋理」の評価関数の値の平均である．定跡が打ち切られた時点での評価値は，歩の価値が 100 点であることを考慮すると，ほぼ互角と言える．勝率と比較すると，序盤の優劣を評価関数で正しく判断することの難しさが示されているといえる．

5. 現在のコンピュータ将棋の問題点とモンテカルロ木探索の利点

実験では，定跡選択では一定の成果をあげたものの，通常探索部分での性能では探索と評価関数によるプログラムには及ばないという結果になった．しかし，探索と評価関数による手法にも，いくつか問題点があることが分かっており，そのような局面ではモンテカルロ木探索が有効である可能性がある．

本章では，2007 年 3 月に行われた渡辺竜王対 Bonanza の対局[14] を例に，現在のコンピュータ将棋が抱える問題点とモンテカルロ木探索の利点について述べる．この対局は，中盤までは互角の形勢に持ち込んでいたものの，終盤に典型的にコンピュータが苦手とする局面に入り，Bonanza が敗れた．

図 4 において，Bonanza は▲ 2 四歩を選択した．しかし，この局面は攻め合いでは先手に勝機はなく，実際この手が敗着となった

この局面の最善手は▲ 2 七香でこれならまだ難しかったとされている[15]．相穴熊のような局面では，正しい局面の評価をするためには，特定の手順について深く読む必要がある．探索と評価関数による手法では，基本的に一定の深さで探索を打ち切るため，このような局面は苦手といえる．

表 7 は，図 4 に今回実装したモンテカルロ木探索に

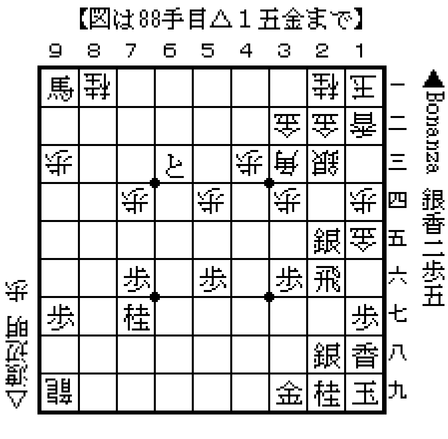

図 4　渡辺竜王 対 Bonanza（局面１）

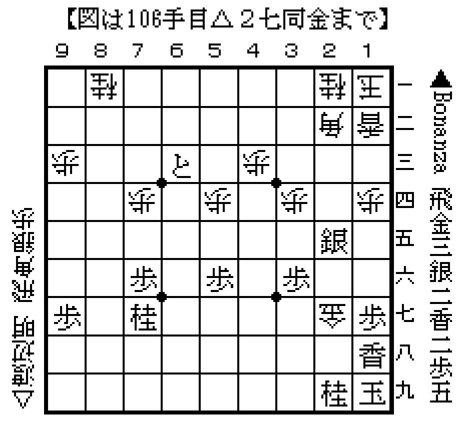

図 5　渡辺竜王 対 Bonanza（局面２）

よるプログラムに解答させたときの結果を示す．なお，評価値はモンテカルロ木探索の勝率を示している．

表 7　モンテカルロ木探索による図 4 の局面の評価

オーダー	指し手	評価値
1（最善手）	▲２七香	0.441
2	▲３七馬	0.407
3	▲３七銀打	0.401
...	...	...

渡辺竜王が最善手として示した▲２七香を選択できていることが分かる．モンテカルロ木探索では，playout が直線的な読みの働きをするため，このような局面では探索と評価関数による手法よりも良い結果が得られることが多い．

図 5 は，すでに敗勢の局面であるが，Bonanza をはじめとした多くのプログラムではほぼ互角の評価をしてしまう[16]．この局面では，先手の持ち駒の多さ，後手玉に王手がかからないことを評価することの難しさなどから，評価関数により正しい形勢判断を行うのが難しい局面といえる．

表 8 は，図 5 の局面を今回実装したモンテカルロ木探索によるプログラムに解答させた結果である．評価値（勝率）が 3 割程度となっており，かなりの劣勢と評価していることが分かる．

表 8　モンテカルロ木探索による図 5 の局面の評価

オーダー	指し手	評価値
1（最善手）	▲２八金	0.332
2	▲３七銀	0.330
3	▲３九銀	0.321
...	...	...

評価関数を設計する際の大きな問題点として，ある程度複雑なゲームでは，局面によって評価すべき項目が大きく異なるということがあげられる．たとえば，図 5 のような局面では，駒割りよりも手番や相手玉に王手がかかるかといったことを重視しなければならない．多くの将棋プログラムでは，局面の進行度により評価関数のパラメータの値を変化させるなどの手法をとっているものの，あらゆる局面に対応できる評価関数の設計は不可能といえる．

モンテカルロ木探索では，駒割りや駒の働き，手番や玉の危険度など多くの項目を個別に評価することはなく，シミュレーションの勝率というゲームの知識に依存しない評価指標を用いるため，より汎用性のある局面の評価が可能である．

6.　今後の展望

本論文では，コンピュータ将棋ではモンテカルロ木探索により従来の手法を単純に上回ることは難しいものの，部分的には十分に有効であることを示した．今後コンピュータ将棋にモンテカルロ木探索を利用する場合，従来の手法と併用し，その利点を生かすことが必要となると考えられる．本章では，モンテカルロ木探索の定跡選択部分における利用，通常探索との併用について今後の展望を述べる．

6.1　モンテカルロ木探索を用いた定跡選択

本論文の結果，定跡選択部分ではモンテカルロ木探索の一定の有効性が示された．

問題点としては，通常の思考部（探索と評価関数）との相性が考慮されていない点があげられる．将棋の序盤は確実な最善手を求めることは不可能であり，現実的にはプレイヤ（プログラム）がより勝ちやすい手順を選択することが目標となる．そのため通常の思考部との相性を考えることは重要であり，大きな課題と言える．

改善策の一つとしては，playout 中の指し手の選択に，プログラムの評価関数の値を用いることが考えられる．この場合，実行時間が問題となるが，定跡選択部のように指し手が絞られている局面では数千〜数万

程度の少ない playout 数でもある程度よい指し手の選択が可能であると考えられる．

また，本論文中では定跡データベース中に存在しない手は全く生成していないが，一致する指し手が少ない局面では，ヒューリスティックにより有力と考えられる指し手を数手生成することなども有効である可能性がある．

6.2 通常探索との併用

モンテカルロ木探索は単純に探索と評価関数による手法を上回ることは難しいと考えられるものの，5 章で示したように，局面によっては非常に有効といえる．モンテカルロ木探索単独で用いるのではなく，従来からの手法である探索と評価関数による手法と併用することは棋力の向上に有効である可能性がある．以下に具体的な例を示す．

- 通常探索の値が安定しないとき，モンテカルロ木探索を利用する
- 通常探索とモンテカルロ木探索を同時に利用し，評価が大きく違う場合には探索深さ，思考時間を延長する

通常探索の値が深く読むほど下がっていくなど，評価値が安定しないことがある．このように探索深さを十分にとることが難しい状況においても，モンテカルロ木探索では，playout が直線的な読みの働きをすることが期待でき，従来の手法より有効に働く可能性がある．

また，現在のコンピュータはマルチコア化が進んでおり，従来の手法とモンテカルロ木探索の両手法を同時に行うことも可能である．全く異なる手法による思考を同時に行い，参考にすることは棋力向上に有効である可能性が高いと考えられる．

7. おわりに

本論文では，モンテカルロ木探索によるコンピュータ将棋を実現した．囲碁において成功した手法に加え，チェスや将棋で古くから用いられているキラームーブやヒストリーヒューリスティックといった考え方を利用することにより，その性能を改善できることを示した．これらの改良により，次の一手問題の正答数では初段程度のプログラムに迫る正答数を得ることに成功した．

有用性という点では，現在のコンピュータ将棋はアマチュアトップに匹敵する強さにまで達しており，モンテカルロ木探索によるプログラムが従来の手法を単純に上回ることは難しいと考えられる．しかし，序盤の定跡選択や一部終盤では，従来の手法を上回る有望な結果を得ることに成功し，コンピュータ将棋においてモンテカルロ木探索の部分的な利用が棋力の向上に有効となり得ることを示した．

現在のコンピュータ将棋は，ほぼすべてのプログラムが同様のロジックに基づいており，抱えている問題点もある程度共通しているといえる．モンテカルロ木探索は，これらの問題点を改善するコンピュータ将棋の新たな実現法として，今後が大いに期待される手法であると考えられる．

参 考 文 献

1) Coulom, R.: Efficient Selectivity and Backup Operators in Monte-Carlo Tree Search, *Proceedings of the 5th International Conference on Computers and Games*, Turin, Italy (2006).

2) 美添一樹：モンテカルロ木探索—コンピュータ囲碁に革命を起こした新手法，情報処理， Vol.49, pp.686–693 (2008).

3) Kocsis, L. and Szepesvári, C.: Bandit Based Monte-Carlo Planning, *Proceedings of the 15th European Conference on Machine Learning(ECML)*, pp.282–293 (2006).

4) Brügmann, B.: Monte Carlo Go, http://ideanest.com/vegos/MonteCarloGo.pdf

5) Coulom, R.: Computing Elo Ratings of Move Patterns in the Game of Go, *In Computer Game Workshop*, Amsterdam, The Netherlands (2007).

6) Gelly, S., Wang, Y., Munos, R. and Teytaud, O.: Modifications of UCT with Patterns in Monte-Carlo Go, Technical Report RR-6062, INRIA (2006).

7) 山下宏：モンテカルロ法の彩について，http://www32.ocn.ne.jp/yss/cgf20080412.html

8) 橋本隼一，橋本剛，長嶋淳：コンピュータ将棋におけるモンテカルロ法の可能性，第 11 回ゲームプログラミングワークショップ，pp.195–198 (2006).

9) 金子知適：兄弟節点の比較に基づく評価関数の調整，第 12 回ゲームプログラミングワークショップ，pp.9–16 (2007).

10) 保木邦仁：局面評価の学習を目指した探索結果の最適制御，第 11 回ゲームプログラミングワークショップ，pp.78–83 (2006).

11) Auer, P., Cesa-Bianchi, N. and Fischer, P.: Finite-time Analysis of the Multiarmed Bandit Problem, *Machine Learning*, Vol. 47, pp. 235–256 (2002).

12) 松原仁：コンピュータ将棋の進歩 2，共立出版 (1998).

13) 将棋タウン棋力判定問題集，http://www.shogitown.com/school/judge/judgetop.html

14) 大和証券杯ネット将棋公式ホームページ，http://www.daiwashogi.net/

15) 渡辺明：渡辺明ブログ，http://blog.goo.ne.jp/kishi-akira/

16) 山下宏：FPGA で将棋プログラムを作ってみるブログ，http://blog.livedoor.jp/yss_fpga/