

5 五将棋における自動対戦を用いた評価関数の学習

塩田 雅弘^{1,a)} 伊藤 毅志^{1,b)}

概要: 本研究では、近年将棋で用いられている評価関数の機械学習の手法を5五将棋に適用することで、5五将棋でもその手法が有効であるかを検証した。具体的には、オープンソースで公開されている「やねうら王」をベースに、この手法を5五将棋に応用したプログラムを開発した。その結果、本プログラムは、2020年3月に開催されたUEC杯5五将棋大会において、全勝優勝した。また、有力プログラムとの対戦結果をもとにその強さを比較した。

1. はじめに

5五将棋は、1970年ごろ考案された現代将棋の派生のゲームである。コンピュータ5五将棋の大会も国内外で行われており、国内では、電気通信大学のエンターテインメントと認知科学研究ステーション主催で、UEC杯5五将棋大会が2007年より開催されている。ここ数年のUEC杯は、大会のレベルは向上している一方で上位プログラムの実力は拮抗している傾向にあり、突出して強いプログラムはまだ登場していない。そのため、プログラムの棋力にはまだ伸び代があると考えている。

一方、近年のコンピュータ将棋ではプログラムの棋力向上が著しく、人間のトップレベルを超えるまでになっている。この要因として、プログラム同士の棋譜を教師とする機械学習手法が成功したことが挙げられる。本研究では本将棋で成功しているこの手法が5五将棋にも有効であるかの検証を行い、既存プログラムよりも高い棋力を有する5五将棋プログラムの作成を試みる。

2. 5五将棋

5五将棋は通常の将棋に比べて盤の大きさが狭く場合の数も少ない、このゲームの初期局面を図1に示す。使用する駒は全部で12枚であり、敵陣1段目のみ駒を成ることができる。UEC杯コンピュータ5五将棋大会では千日手後手勝ち（連続王手の千日手は除く）のルールが採用されており、今日でも多くの大会ではそのルールが用いられている。本研究でもこのルールに準拠するものとする。その

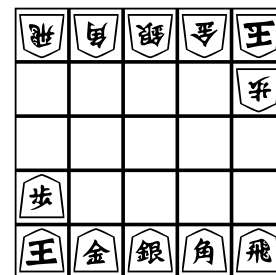


図1: 5五将棋の初期配置
[1]より引用

他のルールについてはほぼ通常の将棋と同様である。

5五将棋のゲームの複雑さをゲーム木探索の大きさと調べてみると、平均合法手を15手程度、平均終了手数を80手程度と見積もれば、10の94乗程度の大きさであることがわかる。これはオセロ（10の60乗程度）よりは大きいがチェス（10の120乗程度）や将棋（10の220乗）よりは小さい程度であることが推定される。もちろん、現在のところ完全解も見つかっていない。

5五将棋は、通常の将棋の特徴を残しつつ、探索量の少ないゲームであるので、将棋では検証に時間がかかる新しい技術を比較的容易にテストできるため、将棋のテストベッドとして、様々な研究が行われてきた。小幡らの研究では、合議アルゴリズムの有効性を初めて検証するのに5五将棋が用いられている [2]。

3. 関連研究

3.1 局面評価の機械学習

評価関数の機械学習を目指した代表的な手法として、保木によって提案されたBonanzaメソッドが挙げられる [3]。棋譜の指し手とMinimaxの一致率を測る目的関数を導入し、棋譜と一致する指し手を選ぶように評価関数の特徴ベ

¹ 電気通信大学
The University of Electro-Communications
a) shioda@minerva.cs.uec.ac.jp
b) taito@mbc.nifty.com

クトルを調整する手法である。保木はプロ棋士の公式戦の棋譜 3 万局と将棋倶楽部 24 の棋譜 3 万局、計 6 万局の棋譜を教師として 1 万を超えるパラメータの自動調整に成功した。

その後、局面の評価値は 3 駒関係と呼ばれる玉を含む 3 駒の位置関係の組み合わせをもって計算するのが主流となり、この場合評価関数のパラメータ数は数千万にも及ぶ。金澤は、膨大な数のパラメータを調整するのに必要な教師データが不足しパラメータが不適切な値を持つことで、プログラムが局面の形勢判断を誤る可能性について言及した [4]。この問題に対して、ある局面を数手先まで探索した評価値を理想の評価値として、この局面から静止探索した局面の評価値をこれに近づける、という強化学習に類似した機械学習法を提案して学習結果の改善を試みた。その結果、この手法を繰り返し適用した Bonzana6.0 のレーティングは 150 程度向上した。

3.2 自己対戦棋譜を利用した学習

近年プログラム同士の棋譜を使って評価関数を学習する手法が確立したことで、コンピュータ将棋プログラムの実力は飛躍的な成長を果たした [5]。プロ棋士とコンピュータの対局では、2013 年にコンピュータがプロ棋士に平手で初めて勝利し 2016 年以降はコンピュータ側の全勝となっている。本項では、現在やねうら王 [6] を始めとする多くの将棋プログラムで使用されている学習手法について説明する。

はじめに、プログラム同士で自己対戦を行い棋譜データを生成する。自己対戦では一定の探索深さ d によって着手を決定するが、一定の確率でランダムに指し手を選択することで学習がローカルミニマムに陥るのを防ぐことを期待する。次に、棋譜データの各局面における評価値を教師としてパラメータの更新を行う。このとき、静止探索の評価値から算出した勝率 q を、教師の評価値から算出した勝率 p に近づけるように学習を行う。シグモイド関数によって評価値を勝率 $(0, 1)$ に変換する。目的関数には 2 値分類の交差エントロピーを使用する。

$$H(p, q) = -p \log q - (1 - p) \log(1 - q) \quad (1)$$

瀧澤は上の教師の評価値を利用した学習と局面の勝敗結果を利用した学習を組み合わせることで、学習がより上手くいくことを示した [5]。このとき目的関数には式 (2) を使用する。

$$H(m, q) = -m \log q - (1 - m) \log(1 - q) \quad (2)$$

$$m = (1 - \lambda)t + \lambda p \quad (3)$$

ここで λ は定数であり、 t は局面の勝敗を表し、勝ちのとき 1、負けのとき 0 をとる。パラメータの更新にはミニバッチ勾配降下法を使用する。以上の自己対戦による棋譜の生成とパラメータの更新を繰り返すことで、プログラムの棋

力向上を図る。

4. 5 五将棋への応用

本研究では、3.2 の学習手法が 5 五将棋にも有効であるかを検証する。プログラムには、オープンソースの将棋プログラムやねうら王 [6] をベースとして、5 五将棋用に変更を加えたものを使用する。やねうら王は 2019 年 5 月に開催された第 29 回世界コンピュータ将棋選手権の優勝プログラムであり、現在最も強力な将棋エンジンの 1 つである。以下、大きくコードを変更した箇所について述べる。

(1) Bitboard の実装

盤面の 1 マスを 1 ビットとして盤面の状態を表す方法を Bitboard と呼ぶ。Bitboard を用いて駒の利きの算出や盤面の更新を高速に処理することで、探索の高速化を図ることができる。通常の将棋の盤面は 81 マスあり、複数の整数を用いて局面を表現するため実装がやや複雑になるが、5 五将棋では 32 ビット整数で盤面状態を表現できるため簡潔に実装できる。

(2) 1 手詰めルーチン

通常の本探索でその局面が 1 手詰めであるかを判定するには、盤面を更新し、合法手が存在しないことを確認する必要がある。1 手詰めルーチンは、盤面の更新をせずに、盤面の状態のみを用いてその局面が 1 手詰めであるかを判定する関数であり、これによって探索の高速化を図ることができる。やねうら王の 1 手詰めルーチンに適宜変更を加えることで、5 五将棋における 1 手詰め局面のおよそ 98% を正しく詰みと判定する関数を実装した。

5. 学習実験

5.1 実験環境

作成した 5 五将棋プログラムを使用して、3.2 の手法を適用し、評価関数の学習を行う。学習ルーチンはやねうら王に実装されているものをそのまま利用する。目的関数は式 (2)、 $\lambda = 0.5$ とする。また動作環境は表 1 の通りである。

5.2 線形モデルの学習

学習する評価関数は、四駒関係 KKP (King-King-Piece-Piece) という、双方の玉を含む 4 駒の位置関係すべてと手番を特徴量を持つ。以下これを KKPPT と呼ぶ。学習する KKPPT のパラメータ数は約 3000 万である。実験の

表 1: 動作環境

CPU	Intel i5-8400 @2.80GHz
OS	Windows 10 64bit
メモリ	8GB

表 2: 各学習における学習前との対戦結果

1000 局, 思考時間 1 手 1 秒		
エポック数	学習前との勝敗	レーティング差
1	988 - 12	766.23
2	730 - 270	172.78
3	555 - 445	38.37
4	529 - 473	19.44
5	511 - 489	7.64

表 3: 探索深さを変化させた場合の対戦結果

1000 局, 思考時間 1 手 1 秒				
	学習前	$d = 3$	$d = 6$	$d = 9$
学習前		488	459	450
$d = 3$	512		476	496
$d = 6$	541	524		464
$d = 9$	550	504	535	

条件は、探索深さ $d = 6$ 、初手から 24 手目までにランダムな指し手 5 回、教師局面数 5 億、ミニバッチサイズ 100 万局面、学習率 30 とした。エポック数は 5 とし、それぞれ学習前の評価関数に対する勝率を測定した。結果を表 2 に示す。1 エポックから 4 エポックでは学習前と比較して有意に勝ち越したが、5 エポックでは学習前の 4 エポックに対して棋力差があるとは認められなかった ($p > 0.05$)。1 エポックでは大幅なレーティング向上が確認されたが、エポック数が増えるにしたがい、レーティングの上昇は徐々に小さくなった。学習を繰り返すと強くなる幅が狭くなることは、パラメータが徐々に正確な値に改善されていることを示していると考えられる。

5.3 自己対戦の探索深さが与える影響

5.2 では、探索深さ d を 6 として自己対戦を行い、その棋譜を教師とした。次に、 d を変化させたときの、勝率の変化について調べる。学習済の KKPPT 評価関数を用いて、探索深さを 3, 6 および 9 とし、それぞれ自己対戦を行い棋譜データからパラメータの更新を行う。教師局面数やミニバッチサイズなどの条件は 5.2 と同様である。エポック数はそれぞれ 1 とした。対戦実験の結果を表 3 に示す。学習前に対する勝率は $d = 9$ の場合が最も高く、 $d = 3$ の場合が最も低くなった。 $d = 3$ の場合は学習前に比べて棋力差があるとは認められなかった ($p > 0.05$)。この原因は、自己対戦棋譜の質が低く、学習効果が低かったためだと予想される。また、表の下段ほど戦績が良いという傾向がみられる。以上より、深い探索により生成された棋譜を教師に用いた方が、より高い学習効果が得られることが示唆された。一般に、探索深さが深いほど探索結果は正確になるため、生成される棋譜の質も向上することが期待される。したがって、この結果は妥当なものであると考える。

表 4: NNUE 評価関数の構造

右側は出力の次元数を表す

入力	9500
全結合	512
Clipped ReLU	512
全結合	32
Clipped ReLU	32
全結合	32
Clipped ReLU	32
評価値	1

表 5: 学習済 KKPPT との対戦結果

300 局, 思考時間 10 分切れ負け		
エポック数	先手番勝敗	後手番勝敗
1	47 - 103	104 - 46
2	50 - 100	144 - 6

5.4 非線形モデルの学習

深層学習の分野では、畳み込みニューラルネットワークとモンテカルロ木探索を組み合わせた AlphaZero の手法が、チェス、囲碁および将棋の 3 つのゲームにおいて非常に強いプログラムを作成できることが示されているが、膨大な量の計算リソースが必要となり、これを模倣するのは困難である。那須らは層の浅いニューラルネットワークを用いた NNUE 評価関数を提案した [7]。この評価関数は CPU で高速に動作する点が特長であり、現在では多くの将棋プログラムで使用されている。本研究で学習する NNUE 評価関数の構造は表 4 の通りである。入力特徴量には、HalfKP と呼ばれる二駒関係の KP を改変したものを使用する。詳細は文献 [7] を参照されたい。

学習手法はこれまで同様、自己対戦棋譜を利用した学習手法を用いる。実験の条件は、探索深さ $d = 12$ 、初手から 24 手目までにランダムな指し手 5 回、教師局面数 1 億、ミニバッチサイズ 1 万、学習率 0.1 とした。エポック数は 2 とした。ただし、0 エポック (初期状態) が生成する棋譜の質は著しく悪いため、学習済の KKPPT で自己対戦させた棋譜を教師データとして、0 エポックからパラメータの更新を行ったものを 1 エポックとしている。学習済 KKPPT を相手に対戦実験を行った結果を表 5 に示す。通算の勝敗を考えると 2 エポックでは KKPPT に有意に勝ち越していることから、棋力は向上していると考えられる。しかし、後手番の勝率は大きく改善された一方で、先手番の勝率にはほとんど変化がなかった。手番による勝率の偏りについては後の章で考察する。

6. 対戦実験

電気通信大学のエンターテインメントと認知科学セッション主催で 2007 年より、コンピュータ 5 五将棋の大会

表 6: KKPPT の対戦結果

300 局, 思考時間 10 分切れ負け		
対戦相手	先手番勝敗	後手番勝敗
Fairy-Stockfish	118 - 32	111 - 39
GA 将?	119 - 31	110 - 40
Shokidoki	115 - 35	147 - 3

表 7: NNUE の対戦結果

300 局, 思考時間 10 分切れ負け		
対戦相手	先手番勝敗	後手番勝敗
Fairy-Stockfish	77 - 73	147 - 3
GA 将?	130 - 20	104 - 46
Shokidoki	127 - 23	150 - 0

が開催されている。2020年3月15日には第12回大会開催され、8プログラムが参加した。大会は、先手後手1局ずつの総当たり、思考時間は10分切れ負けで実施された。筆者は、本研究で作成したプログラムと学習済のNNUE評価関数で本大会に出場し、14局全てで勝利した。大会の結果は付録A.1に記載した。

本研究で作成したプログラムが、既存のプログラムに比べて十分に強いプログラムであるかを評価するには、多くのプログラムと対戦する必要があると考える。そこで、学習済の2つの評価関数KKPPTとNNUEを用いて、それぞれUEC杯の上位プログラムとの間で先後150局ずつ合計300局の対戦実験を行った。結果を表6と表7に示す。どちらの評価関数も3つのプログラムすべてに有意に勝ち越す結果が得られた。表7に注目すると、Fairy-StockfishとShokidokiに対する後手番の勝率は非常に高くなった一方でGA将に対する後手番の勝率は先手番よりも悪くなっている。またFairy-Stockfishに対する先手番の勝率は5割程度であり、これはKKPPTに比べて低い勝率となった。これらは単にプログラム間の相性の問題であるとも考えられるが、棋譜を丹念に調べ同様の局面が多数現れていなかったなど、その内容を精査する必要があると考えている。

7. 手番による勝率の偏りに関する考察

これまでの対戦実験の結果を考慮すると、5五将棋は後手が勝ちやすいゲームであることが推察される。これは「千日手後手勝ち」というルールが影響を与えていることが考えられる。そこで手番による有利不利を調査するため、学習済のNNUE評価関数を用いて自己対戦（思考時間10分切れ負け）を行った。その結果、先手9勝・後手291勝と大きく後手番が勝ち越すこととなった。図2は、この自己対戦において頻繁に確認された局面である。このゲームは、局面において有力な手（形勢が明らかに不利にならない指し手）は千日手を避ける手以外は非常に少ないため、局面が膠着しやすいのだと考えられる。局面が手詰まりに

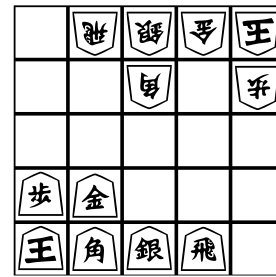
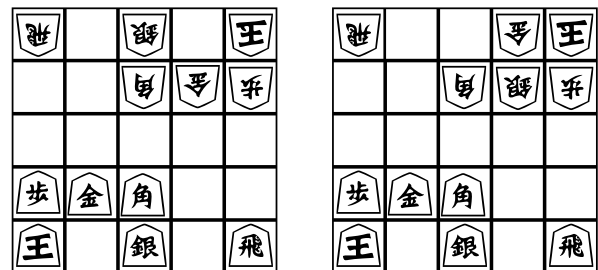


図 2: 8 手目△2一金まで



(a) △2二金まで

(b) △2二銀まで

図 3: 初手から4手進めた局面

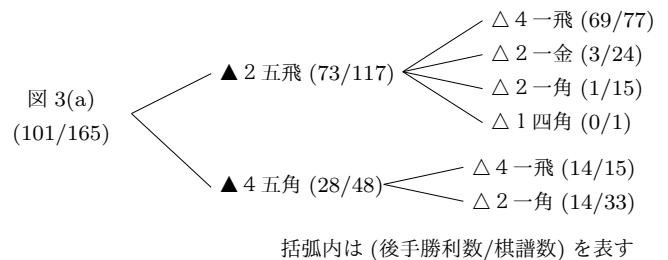


図 4: 4 手目以降の変化とその勝敗

なったとき、後手側からは積極的に動く必要はないが、先手側は千日手を避けるために局面を打開しなければならない。実際に図2以降、局面が膠着し先手が無理に打開して惨敗する棋譜が多く見られた。このうち2つの棋譜を付録A.2に紹介する。

NNUE 同士の自己対戦 300 局のうち、221 局は初手から8手進めた局面が図2に一致（手順前後を含む）しており、類似した棋譜が多く、同様の局面が多く現れていることが確認される。これは同種のプログラムであるため、同様の局面を学習した影響が一因として考えられる。そこで、異種プログラムであるFairy-Stockfishを用いて同じ条件で自己対戦を行った。その結果、自己対戦の結果は先手170勝・後手130勝であった。300局のうち165局は初手から4手進めた局面が図3(a)に一致し、手番ごとの勝敗は先手64勝・後手101勝となった。また300局中106局で初手から4手進めた局面が図3(b)に一致し、この場合は先手89勝・後手17勝となった。図4は棋譜で確認された、図3(a)以降の指し手の変化と後手の勝率を示したものであり、どちらの変化でも△4一飛を選択したとき後手の勝率が大き

く上昇している。図3と図4どちらも後手が自身の期待勝率を高くする変化を選択することが可能である。また図2の局面は83局で確認され、後手が74勝と大きく後手が勝ち越している。Fairy-Stockfish 同士の棋譜は NNUE 同士の棋譜に比べて内容の偏りが少ないが、却って後手が損をしている棋譜が多く含まれていると考えられる。後手の指し手の選択次第では、後手の勝率の方が先手の勝率を上回ることが可能となる。

本研究で作成したプログラムと Fairy-Stockfish は全く異なる評価関数で動作している。しかしそのどちらの自己対戦棋譜からもこのゲームは後手有利となる可能性が示唆された。

8. まとめと課題

本研究では、近年のコンピュータ将棋で成功を収めている評価関数の機械学習手法を5五将棋に適用した。第29回世界コンピュータ将棋選手権の優勝プログラムやねうら王をベースとする5五将棋プログラムを作成して学習を行った。評価関数は四駒関係と非線形モデルの2種類を学習し、そのどちらでも既存のプログラムに勝率7割以上を記録した。これにより、学習手法が5五将棋にも有効であることを示した。同時に、自己対戦時の探索深さが深いほど学習がうまくいくことを確かめた。また自己対戦の棋譜から手番による勝率の偏りについても調査し、開始数手の指し手が勝敗に大きく影響していることを示した。

今後の課題として、棋力の評価方法を工夫する必要がある。同じ対局条件で自己対戦を繰り返す場合、同じ指し手を選ぶ可能性が高くなるため、生成される棋譜に偏りが生じてしまうことは避けられない。持ち時間やスレッド数を調整し、棋譜のバリエーションを増やす必要がある。またプログラムの棋力向上も課題である。通常の将棋における平均合法手は約80手とされており、本研究でベースとして使用したやねうら王の探索部でも大幅な枝刈りが実装されている。5五将棋は通常の将棋に比べて探索空間がはるかに狭いゲームであるため、枝刈りの調整は棋力向上に寄与すると考えている。

参考文献

- [1] 5五将棋 portal : <http://minerva.cs.uec.ac.jp/cgi-bin/uec55shogi/wiki.cgi> (参照 2020.06.01)
- [2] 小幡拓弥, 埴雅織, 伊藤毅志, 思考ゲームによる合議アルゴリズム~単純多数決の有効性について~ : GI22, 2号, pp.1-5(2009)
- [3] 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, ゲームプログラミングワークショップ 2006 論文集, pp.78-83(2006)
- [4] 金澤裕治 : 教師データが不足した環境での機械学習結果改善手法, 情報処理学会論文誌, 57巻, 11号, pp.2382-2391(2016)
- [5] 瀧澤誠, 伊藤毅志 : 進化し続けるコンピュータ将棋 2 elmo

- の開発と技術—第27回世界コンピュータ将棋選手権優勝プログラムインタビューから, 情報処理, 59巻, 2号, pp.153-159(2018)
- [6] やねうら王 : (<https://github.com/yaneurao/YaneuraOu>) (参照 2020.06.01)
 - [7] 那須悠 : 高速に差分計算可能なニューラルネットワーク型将棋評価関数, 第28回世界コンピュータ将棋選手権アペール文書 (https://www.apply.computer-shogi.org/wsc28/appeal/the_end_of_genesis_T.N.K.evolution_turbo_type_D/nnue.pdf) (参照 2020.06.01)

謝辞 本研究は JSPS 科研費 18H03347 の助成を受けたものです。

付 録

A.1 第12回 UEC 杯5五将棋大会結果

順位	プログラム名	開発者 (代表)	勝敗
1	ShioRamen	塩田 雅弘	14 - 0
2	Fairy-Stockfish	Fabian Fichter	9 - 5
3	GA 将?	森岡 祐一	8 - 6
3	Shokidoki	H.G.Muller	8 - 6
5	Lima v6.0	Ferdinand Mosca	7 - 7
6	erweitern_55	中屋敷 太一	6 - 8
7	NDHU Polygames	Shi-Jim Yen	3 - 11
8	Go!Go!あうあう将棋	氏家 一朗	1 - 13

A.2 自己対戦棋譜

手合割：5五将棋

▲4四金	△3二角	▲3四角	△2二金
▲2五飛	△4一飛	▲4五角	△2一金 (図2)
▲3三金	△2二銀	▲同 金	△同 金
▲4四銀打	△2一金打	▲3四銀	△3一金
▲3三銀直	△同 金	▲同 銀	△2二銀
▲4四銀	△2一角	▲2四金	△3二角
▲3四金	△2一角	▲5三步	△3二角
▲1五飛	△2一角	▲3五飛	△3二角
▲2五飛	△1四角	▲2四飛	△3二角
▲3五金	△1三銀	▲2五飛	△2二銀
▲1五飛	△2一角	▲3四金	△3二金
▲5四角	△2三銀	▲3五金	△2二金
▲2一角成	△同 玉	▲5二角	△3一飛
▲2五飛	△1一玉	▲4五金	△4二角
▲3五飛	△3二金	▲5四玉	△5一飛
▲3四金	△5二飛	▲同 歩	△3四銀
▲4一飛	△3一金打	▲同飛成	△同 角
▲3四飛	△2一角	▲3五金	△2二角
▲3二飛	△同 角	▲4三銀打	△同 角
▲同 玉	△4一飛	▲4二金	△同 飛
▲同 玉	△4一飛	▲3二玉	△3一金
▲2三玉	△4四飛	▲同 金	△同 角
▲2一飛	△同 金	▲3三角	△2二銀
▲同 角	△同 金	▲1四玉	△2三銀
▲2五玉	△3五飛	まで98手で後手の勝ち	

手合割：5五将棋

▲3四角	△2二金	▲4四金	△3二角
▲4五角	△4一飛	▲2五飛	△2一金 (図2)
▲2四銀	△2二金	▲3五飛	△4二銀
▲1五飛	△2一玉	▲3四角	△1一玉
▲4五飛	△5一飛	▲2五飛	△1四角
▲1五飛	△3二角	▲4五角	△4一飛
▲3五飛	△1四角	▲3四角	△3二角
▲2五飛	△5一飛	▲4五角	△3一飛
▲3五飛	△2一飛	▲3四金	△4一飛
▲1五飛	△2一角	▲2三銀	△3三銀
▲2二銀	△同 銀	▲2五飛	△3三銀打
▲2四金打	△3四銀	▲同 金	△3三金
▲3五金	△3二角	▲3四銀	△同 金
▲同 金	△3三銀打	▲3五金打	△4三角
▲同 金	△同 飛	▲5三角	△4二金
▲2二飛	△同 銀	▲4四銀	△3一飛
▲4二角	△同 飛	▲2四金打	△1五角
▲3四金寄	△4一飛引	▲5三步	△4二角成
▲5四角	△1五馬	▲4五金	△1四馬
▲5二歩	△1三馬	▲5三銀	△2三銀
▲4四金寄	△2二馬	▲5一歩成	△同 飛
▲4二銀	△5四飛	▲同 玉	△2一角
▲5三玉	△4一歩	▲5一飛	△4二歩
▲3一飛成	△同 馬	▲5一飛	△4一飛
▲同飛成	△同 馬	▲3三金	△2四銀打
▲4四玉	△3三銀	▲5五玉	△4三步
▲5四飛	△1五飛	▲2四飛	△同銀直
▲5四玉	△5二飛	まで110手で後手の勝ち	