

局面の情報を利用した，静止探索中の動的な手生成

竹内 聖悟[†] 金子 知適[†] 山口 和紀[†]

本研究では，局面の情報を利用した，静止探索中の動的な手生成を目的とする．静止探索は，探索末端において評価を安定させるために導入された選択的探索である．駒を取る手や王手が生成されるが，どんな手を生成すると良いかはわかっていない．本稿では，将棋において動的な手生成手法の例として詰める予測関数を利用した王手生成手法を作成し，その有効性を示した．また，局面によって生成する指し手を変えることの根拠として，全幅探索と，他の手生成手法との比較に指し手の特徴についての調査をおこない，局面の進行度によって有効な手が異なることなどを示すことができた．

Dynamic Move Generation in Quiescence Search, Using Information of Game Positions

SHOGO TAKEUCHI,[†] TOMOYUKI KANEKO[†] and KAZUNORI YAMAGUCHI[†]

This paper presents a framework for a dynamic move generation in quiescence search, using information of game positions.

In order to obtain stable evaluation, quiescence search is introduced in chess-like games. Generally, capture moves and check moves are generated, however, it is not known that what kind of move should be generated.

In this paper, we prepare check move generator, using predictor of threatmate, as an example of dynamic move generator in Shogi. Our experiments showed the effectiveness of dynamic move generation in quiescence search, and showed that the effective moves in quiescence search are different by game phase.

1. はじめに

強いゲームプログラムの作成には，探索と評価関数が重要である．正確な評価，効率的な探索を行うことができれば，より強いゲームプログラム作成することができる．探索の末端局面は評価関数による評価値がつけられ，評価値に従って最善手が選択される．チェスや将棋などのゲームにおいては，駒の取り合いや成る手などによって評価値に大きな変動が起こるため，探索末端において評価関数を呼ぶのではなく，静止探索と呼ばれる選択的探索を行い，より正確な評価を得ることが行われている¹⁾．

静止探索は局面の正確な評価のために行われるが，どのような手を生成するかは開発者が各自決めている．一般的に，評価値に大きな影響を持つ，取る手，成る手が生成されるが，王を動かす手や王手，取られそうな駒を逃す手などを生成するプログラムもあり，手生成についての明確な基準はない．また，将棋にお

いては，序盤では駒得が，終盤では攻めの速度が重視されるなど，評価関数の重要な特徴はゲームの進行度によっても異なる．よって，静止探索における有効な指し手もまた進行度によって変わると考えるのが自然であるが，従来の静止探索では局面に応じた手生成はあまり行われていない．

本研究では，これらの問題を解決するため，局面の状況に応じて，動的に適切な手生成を提案する．本稿では，動的な手生成手法の例として，詰める予測関数を利用した王手生成手法を採用し，全王手を生成する手法よりも有効であることを示した．また，全幅探索とその他の静止探索との比較，調査を行い，進行度に応じて手生成を変えることの有効性を示した．

一般に，静止探索は探索の中で大きな割合を占めており，静止探索の改善によりプログラムの強さが改善することが期待される．

本稿の構成は以下の通りである．まず，2章で静止探索について述べる．3章で提案手法について説明し，4章で提案手法の評価実験，静止探索における手生成に関する調査について結果を示す．5章では結論を述べる．

[†] 東京大学大学院総合文化研究科
Department of General Systems Studies, Graduate School of Arts
and Sciences, The University of Tokyo
{takeuchi,kaneko,yamaguchi}@graco.c.u-tokyo.ac.jp

2. 静止探索

ゲームプログラムは、探索の末端局面に評価関数による評価値をつけ、評価値に従って最善手を選択する。チェスや将棋などのゲームにおいては、駒の取り合いや成る手などによって評価値に大きな変動が起るため、探索末端において評価関数を呼ぶのではなく、静止探索と呼ばれる選択的探索を行い、より正確な評価を得ることが行われている¹⁾。

静止探索は局面の正確な評価のために行われるが、正確な評価の統一的な定義はなく、静止探索の実装は開発者が各自で決めている。

一般的に静止探索中では、評価値に大きな影響を持つ、取る手、成る手が生成されているが、王を動かす手や王手、取られそうな駒を逃す手などを生成するプログラムもあり、手生成についての明確な基準はない。

静止探索と通常探索との違いとして、探索を行わずに評価値をそのまま利用する Stand Pat がある。今、静止探索では駒を取る手を読むとし、現局面は取り返されて駒損するような手しかないとする、駒を取る手を読んだ結果として駒損した評価値を返すことになり、取る必要のない駒を取ったことにより不正確な評価値となってしまう。しかし、手を読まずに評価関数を利用する Stand Pat により駒損のない評価値を返すことができ、正確な評価が可能となる。

2.1 将棋プログラムにおける静止探索

将棋における静止探索は、各プログラム毎に実装が異なる。

Bonanza は、駒損しない駒を取る手、成る手、王の移動、一手詰の王手を生成している¹¹⁾。

KFEnd³⁾ では、取られそうな駒を考慮した静止探索を行っている。

YSS⁵⁾ では探索の末端付近において生成する指し手に制限をかけ、末端において2手の駒取り探索を行うなど、静止探索相当のことを行っている。また、探索末端の評価では、評価値に取れる駒の価値を足したものを利用するなど、駒を取ることを考慮した評価を行っている。

Tacos⁶⁾ も YSS 同様に、探索末端において取り合い静止探索は行わず、探索末端の評価において、評価値に取れる最大の駒の価値の75%を足したものを利用している。なお、Tacos は実現確率探索⁴⁾を利用して、探索の末端付近では駒損手を生成しない工夫がされている。

2.2 関連研究

将棋における静止探索は様々な実装が見られるが、

静止探索について研究された例は少ない。

筆者らは、将棋において前向き枝刈手法である ProbCut²⁾ を静止探索へ応用する研究を行った⁸⁾。ProbCut は浅い探索による評価を用いて、深い探索による評価を推測し、枝刈を行う手法である。

この中で、静止探索の探索深さについてデータが取られており、探索深さ4,8のペアについて評価値の相関と分散をゲームの進行具合別に計算した結果、終盤ほど分散が大きくなる傾向があることが分かっている。しかし、その研究の中ではゲームの進行具合について考慮した実験は行われていない。

3. 提案手法

本稿では、静止探索において局面の状況に応じた、動的な手生成を提案する。

将棋においては、序盤では駒得が、終盤では攻めの速度が重視されるなど、評価関数の重要な特徴はゲームの進行度によって異なる。そのため、静止探索における有効な指し手もまた進行度によって変わると考えるのが自然であり、局面の状況に応じて動的な手生成が必要となる。

本稿では、局面の状況に応じて動的な手生成をする関数の例として、王手を動的に生成する関数を作成した。まず、詰めろを判定する評価関数⁷⁾を作成した。手生成の関数は、2手前の指し手が詰めると判定された場合に王手を生成する関数とした。これは、こちらの詰めろを相手が受けなかった時に、詰みがあるか王手を読むという意図である。この関数により、王手の生成は詰みの多い局面に限定され、全王手を生成する場合よりも探索ノード数が減少し、効率的な探索が可能となる。

3.1 詰めろ評価関数

実験で用いた詰めろ評価関数について説明する。教師には、棋譜中に現れる詰めろを正例として、詰めろの見つかった局面の詰めろでない合法手、同じ棋譜中に現れた詰めろでない指し手を負例として用いた。特徴は、指し手が駒を取ったか、敵玉の八近傍に利きをつけたか、敵玉は危険か、などを用いて、最尤法によって詰めろ評価関数を作成した。

作成した詰めろ評価関数の性能を測るため、テストデータとして詰めろ9,144手、非詰めろ18,288手の合計27,432手を用意し、評価関数によって分類を行った。結果を表1にまとめた。

4. 実験

将棋を対象として、提案手法の有効性を示す実験を

Accuracy	True Positive	False Negative
0.8804	0.7982	0.9215

行った。また、動的な手生成の妥当性を示すため、局面の進行度によって有用な指し手が変わるか調べた。指し手の特徴について静止探索に用いられている指し手を分類し、駒を取る手、王手などを様々な手を生成する静止探索を全幅探索と比較し、評価値の差の平均、分散、探索ノード数を調べ、進行度別にも調べることで、有用な指し手の特徴について調査した。

これらの実験は、全幅探索との比較による評価となっている。この評価法の妥当性を示すため、各手法を静止探索として利用し、問題集を解かせ、正答数の比較を行った。

実験に用いたプログラムは GPS 将棋 (ver.3764, Open Shogi Library ver.2033) である。

4.1 全幅探索との比較による評価

まず、全幅探索と近いものほど良いと仮定し、全幅探索に近くなるような指し手を調べることを考えた。ある指し手を読んだ時に全幅探索と近いことの指標として、全幅探索の評価値との差の分散を考える。分散が小さいほど全幅に近く、良い結果となる。

評価値の差の分散を求めるために、棋譜 1,000 局を対象として 30 手目以降の局面で深さ 5 の探索を行い、全幅探索の結果との差を取り、平均、分散を求めた。

全幅探索 (All)、探索せず評価値を返す (Eval) の他に、静止探索中に生成する手の候補として以下を用意した。

- Eval: 何も読まずに評価値を返す
- SEE: SEE が正の値で、駒を取る手
- Capture: 駒を取る手
- King: 王を移動する手
- Escape: 利きのついている駒を移動する手
- Promote: 駒を成る手
- SKEP: SEE, King, Escape, Promote の組み合わせ
- Check: 王手
- Threatmate: 提案手法。詰める判定関数により、2 手前の手が詰めると判定されていれば王手

また、実際は SEE や Capture と他の手を組み合わせたものについて複数データを取っている。なお、SEE とは Stataic Exchange Evaluation の略で静的駒交換値のことである。

program	average	std.dev	#nodes
Check	-1,467.41	182,957.99	674,805
Threatmate	-1,286.51	190,128.08	119,078
SEE, Check	-140.65	87,127.92	2,015,238
SEE, Threatmate	-70.29	89,809.63	1,456,588

program	average	std.dev	#nodes
Eval	-2,433.47	199,156.38	83,631
SEE	-27.73	91,671.75	729,561
Capture	490.43	77,737.00	2,580,038
King	361.77	179,778.89	411,554
Escape	1,549.51	140,139.31	7,248,201
Promote	-3,717.29	228,665.21	595,642
SKEP	770.57	58,271.28	24,671,407
SEE, Escape	331.58	70,370.71	12,759,303
Check	-1,467.41	182,957.99	674,805
Threatmate	-1,286.51	190,128.08	119,078
All	-	-	976,379,285

4.1.1 提案手法の有効性

提案手法の有効性を確認するため、Check と Threatmate との比較を行う。結果を表 2 に示す。それぞれ単独のもの比較すると、Threatmate の方が探索ノード数は小さいが、分散が大きくなっており、評価は難しい。一方、SEE との組み合わせを見ると、探索ノード数は約 2/3 で Check との分散の差は小さく、Threatmate による動的な手生成が有効であることが確認できた。

4.1.2 指し手の特徴についての調査

代表的な結果を表 3 に示す。全幅探索の探索ノード数は 976,379,285 であり、全局面数は 83,631 であった。もっとも多くの手を読む SKEP が一番分散が小さく、全幅に近いことが分かる。Capture, SEE が次に分散が小さく、一般に駒を取る手が有効であるという事実が確認された。Escape は探索ノード数が大きい一方で分散があまり小さくなっておらず、静止探索において単独で用いるのには向いていないと考えられる。しかし、SEE との組み合わせの結果を見ると探索コストは上がるが分散も小さくなっており、駒を取る手と逃げる手のように指し手の組み合わせが重要であることがわかる。

続いて、序盤と終盤とでは静止探索に有効な指し手が違うという仮定について調べるため、序盤とそれ以外に分けて分散を求めたところ、表 4 のような結果となった。多くは序盤では分散が大きく、残りでは小さくなる傾向がある。ただし、promote はその差が大きく、序盤よりも終盤で有効であると考えられる他、王を動かす手は序盤で分散が小さくなる傾向があり、終盤よりも序盤で有効であると考えられる。

<http://gps.tanaka.ecc.u-tokyo.ac.jp/gpsshogi/>
<http://gps.tanaka.ecc.u-tokyo.ac.jp/gpsshogi/pukiwiki.php?OpenShogiLib>

program	total	opening	rest
Eval	189,382.12	220,138.41	184,151.29
SEE	101,149.59	84,969.95	64,248.63
King	178,104.66	193,898.38	149,782.49
Escape	151,701.28	134,384.09	101,810.33
Promote	260,466.35	200,792.88	142,326.39
SKEP	63,601.63	55,726.25	39,717.09
SEE, Escape	77,483.85	64,984.82	51,041.61

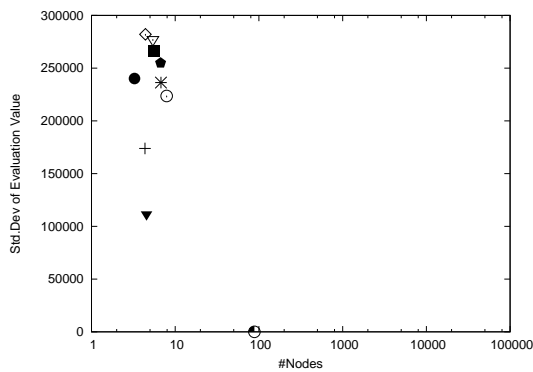


図 1 静止探索 d=1 (探索ノード数:評価値の差の分散)

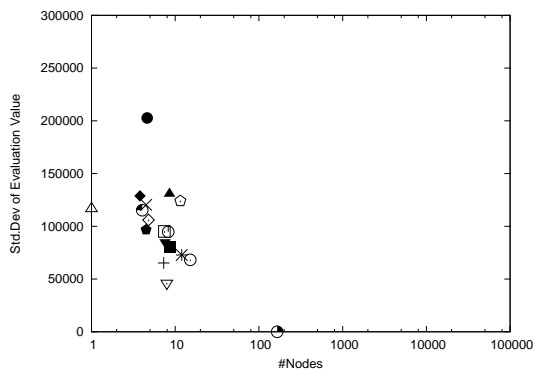


図 2 静止探索 d=2 (探索ノード数:評価値の差の分散)

4.1.3 探索ノード数と評価値の差の分散の関係

分散と探索ノード数の関係を見るために、 x 軸に探索ノード数を、 y 軸に分散をプロットした。なお、 x 軸は対数をとっている。

探索深さを 1 から 5 の間で変えた結果が図 1, 2, 3, 4, 5 である。 $x = 1$ 上にある点は Eval で、 $y = 0$ 上にある点が All である。

複数組み合わせると、当然ながら探索ノード数が増加、分散が減少する。

追加する手の種類によって様々だが、追加するほど探索ノード数が増加し、分散が小さくなり、SEE, Capture に手を加えていくと、プロットした点がほぼ直線

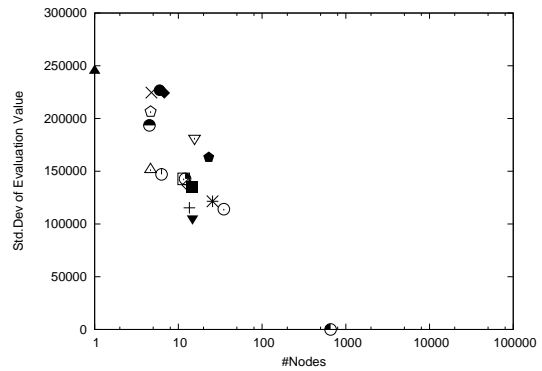


図 3 静止探索 d=3 (探索ノード数:評価値の差の分散)

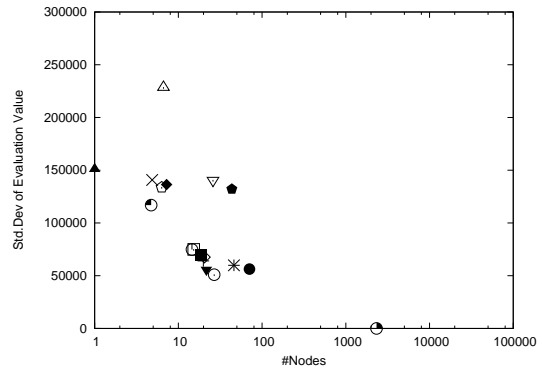


図 4 静止探索 d=4 (探索ノード数:評価値の差の分散)

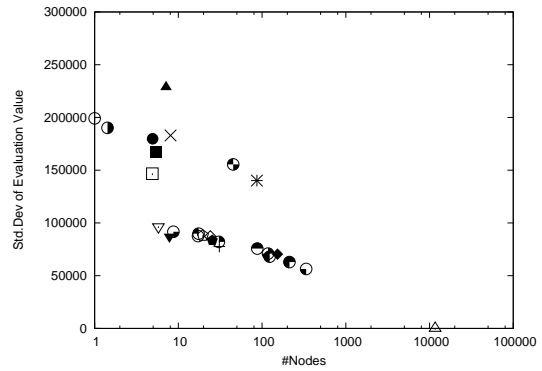


図 5 静止探索 d=5 (探索ノード数:評価値の差の分散)

になっている。これは特に図 5 において顕著である。

次に、探索深さ 5 の全幅探索の結果を序盤、中盤、終盤に分けてプロットした。結果は 6, 7, 8 である。なお、序盤: 進行度 4 以下, 中盤: 進行度 5 以上 9 以下, 終盤: 進行度 10 以上とした。全体及び序盤では Promote は性能が悪かったが、ゲームが進むほど性能が向上しており、中盤、終盤においては有効であることがわかる。

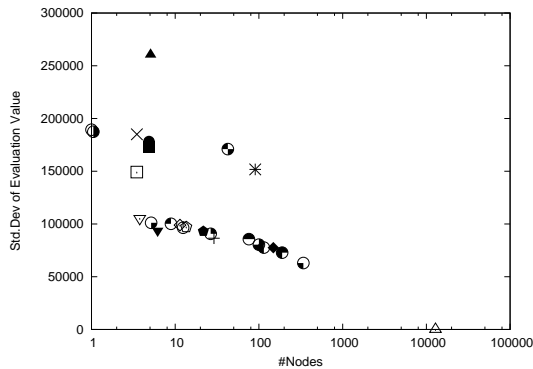


図 6 静止探索 序盤 (探索ノード数:評価値の差の分散)

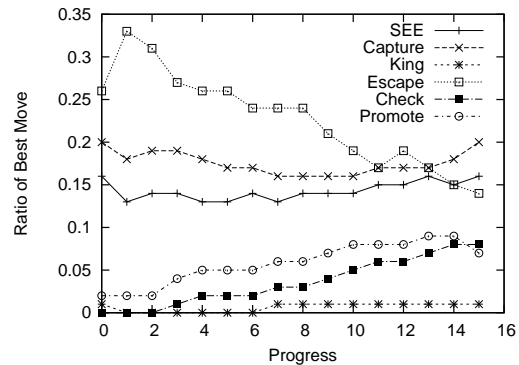


図 9 最善手

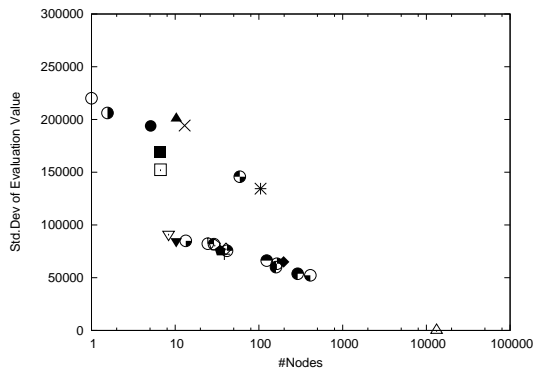


図 7 静止探索 中盤 (探索ノード数:評価値の差の分散)

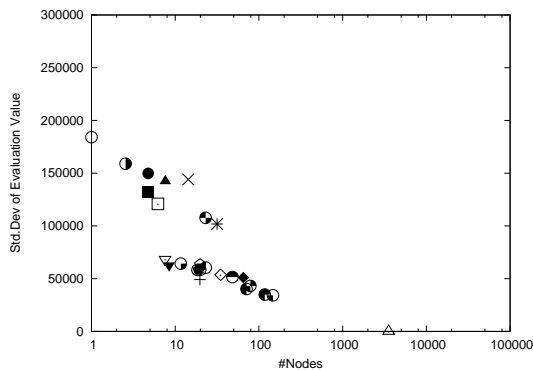


図 8 静止探索 終盤 (探索ノード数:評価値の差の分散)

図 9 から、終盤に近くなる、つまり進行度が大きくなるほど、Escape が減少し、Check, Promote が増加していることが分かる。これから、ゲームの進行度によって適切な手が異なることが分かる。

4.2 全幅探索の性能

これまでの実験は、全幅探索の結果と近いほど良いという仮定が置かれている。ここでは、その仮定の正当性を示すための実験を行う。

全幅探索やその他の探索の静止探索としての性能を測るため、各手法を静止探索として利用し、問題集^{9), 10)}を解かせ、結果を表 5 にまとめた。今回静止探索として、全幅探索, SEE, SKEP (SEE, King, Escape, Promote), Rating (上位 20%) を利用した (Rating は指し手の実現確率のようなもの、値が高いほど良い)。比較のため、全幅探索で探索が既定の深さまで達した問題のみ掲載しており、全問題数は 300 問である。なお、静止探索部分は深さ 4 で探索延長は行われぬ。表中の #qnodes は静止探索の探索ノード数を表す。

結果を見ると、全幅探索より有意に正答したものは無い。また、SEE と SKEP を比較すると、前の実験において全幅探索との分散の小さかった SKEP の方が SEE よりも正答数は多くなった。問題集の結果と全幅探索との比較による評価が一致しており、前の実験で全幅探索との近さを指標としたことの妥当性が示せたと考えられる。

最後に、Rating 上位 20%の手を選んだ結果を見ると、正答数は有意な差はないが最大で、探索ノード数は SKEP と同等となった。Rating は局面に応じて生成する手、およびオーダーが変わるので他のものよりも良い結果が期待されたが、有意な差とはならなかった。Rating は実際の棋譜に現れる局面で有効なもので、静止探索において有意に良い結果が出なかったことは、実際の局面において読むべき手と静止探索において読

続いて、実際に最善手として選ばれる手がどのカテゴリに属するかを調べた。探索深さ 5 の全幅探索を 100 局の棋譜に対して行い、その中で最善手となった手がどのカテゴリに属するかを調べた。各カテゴリについて、 x 軸に進行度、 y 軸に最善手の割合をプロットした結果が 9 である。なお、複数のカテゴリに属する手もあり、正確には割合ではない。

表 5 問題集

program	#correct	#qnodes
All	231	3,560,470,971
SEE	217	37,584,266
SKEP	234	208,309,850
Rating (top 20%)	242	208,348,181

むべき手が異なるというアイデアの傍証となると考えられる。

5. おわりに

本稿では、静止探索において局面の状況に応じて動的に適切な手生成を行う関数の作成を提案した。従来、静止探索での手生成については明確な基準はなく、また進行度などゲームの状況によって手生成を変えることはされてこなかった。

動的な手生成を行う関数の例として、詰める予測関数を利用した王手生成関数を作成した。詰めるの後続でのみ王手を読むことで、全王手を読むよりも効率的な探索が可能になると考えられる。将棋を対象として実験を行い、提案手法の有効性を示した。また、局面の状況、特に進行度によって有効な指し手が異なることを示した。

今後は、進行度や手番、評価値、残り深さの偶奇など局面の情報をさらに集め、それらのデータを元に、より有効な手生成関数の作成を行う。最終的に、作成した手生成の関数を実際のプログラムへ応用し、問題集や自己対戦などから本手法の有効性を示す。

参 考 文 献

- 1) D. F. Beal. A generalised quiescence search algorithm. *artint*, 43:85–98, 1990.
- 2) M. Buro. Probcut : An effective selective extension of the $\alpha - \beta$ algorithm. *ICCA Journal*, 18(2):71–76, 1995.
- 3) 有岡. 将棋プログラム KFEEnd における探索. 松原 (編), コンピュータ将棋の進歩 4, pp. 18–40. 共立出版, 2003.
- 4) 鶴岡, 横山, 丸山, 近山. 局面の実現確率に基づくゲーム木探索アルゴリズム. 第 7 回ゲームプログラミング ワークショップ, pp. –, 2001.
- 5) 山下. Yss-「コンピュータ将棋の進歩 2」以降の改良. 松原 (編), コンピュータ将棋の進歩 5, 第 1 章, pp. 1–32. 共立出版, 2005.
- 6) 橋本. 将棋プログラム tacos のアルゴリズム. 松原 (編), コンピュータ将棋の進歩 5, 第 2 章, pp. 33–67. 共立出版, 2005.
- 7) 竹内, 金子, 川合. 棋譜に基づく詰めるの指標. 第 9 回ゲームプログラミング ワークショップ, pp. 48–53, Nov. 2004.

- 8) 竹内, 金子, 川合. 将棋における probcut の静止探索への応用. 情報処理学会研究会報告, 2005-GI-14, pp. 9–15, 2005.
- 9) 日本将棋連盟書籍 (編). ラクラク次の一手 基本手筋集. 日本将棋連盟, 2003.
- 10) 日本将棋連盟書籍 (編). ラクラク次の一手 2 基本手筋集. 日本将棋連盟, 2003.
- 11) 保木. コンピュータ将棋における全幅探索と futility pruning の応用. 情報処理, 47(8):884–889, 2006. ミニ小特集 コンピュータ将棋の新しい動き.