

将棋における合議アルゴリズム： 既存プログラムを組み合わせて強いプレイヤーを作れるか？

小幡拓弥[†] 杉山卓弥^{††} 保木邦仁^{†††} 伊藤毅志[†]

概要

本稿では、将棋プログラムの新たな並列処理手法である”合議アルゴリズム”を提案し、その効果について検証した様々な実験を報告する。合議アルゴリズムは複数の思考プログラムの意見をまとめて指し手を決定する手法で、疎結合マルチプロセッサにおける動作によく向いている。筆者らは合議アルゴリズムを用いて強豪プログラム同士を連結し、より強いプログラムを作ることに成功した。また、単一の思考プログラムから簡易な方法で複数のプレイヤーを作り合議させる手法を提案し、その有効性を示した。

Consultation Algorithm in Shogi: Can a set of players create a single strong player?

Takuya Obata[†], Takuya Sugiyama^{††}, Kunihito Hoki^{†††}, Takeshi Ito[†]

Abstract

A new algorithm which runs on computer hardware consisting of interconnected processors is designed in Shogi. The algorithm adopts consultation with many individual players. Furthermore, a method which creates multiple players from one is presented. Applying a simple rule to select a single move decision, the consultation algorithm improves the performance of computer Shogi engines. It is also shown that a council system consisting of three famous Shogi programs: YSS, GPS and Bonanza, plays better than each of the three programs.

[†] 電気通信大学情報工学科

Department of Computer Science, University of Electro-Communications

^{††} 慶應義塾大学大学院理工学研究科

Graduate School of Science and Technology, Keio University

^{†††} 東北大学大学院理学研究科

Graduate School of Science, Tohoku University

1. はじめに

1965年に、集積回路におけるトランジスタの集積密度が2年ごとに倍になるという経験則がゴードン・ムーアにより提唱された¹⁾。以来、トランジスタの集積度は順調に高くなるが、2000年台後半からプロセッサ単体の性能は伸び悩んでいる。

近年、ソフトウェアの更なる性能向上のために、様々な分野で並列化の試みが盛んに研究されている。コンピュータ将棋においても効率の良い並列処理は重要な課題であるが、思考アルゴリズムの要であるMinimax法に基づくゲーム木探索を並列化することは容易ではない。このゲーム木探索をアルファ・ベータ法に基づき密結合並列処理するアルゴリズムとしてPrincipal Variation Splitting (PVS)及びDynamic Tree Splitting (DTS)法が提案されており、これらの手法は将棋においても一定の成功は収めている²⁾。

本研究では、クラスタ等の疎結合並列処理を前提とした合議アルゴリズムという手法を提案する。合議アルゴリズムは、複数プログラムが別々に挙げる候補手から一定のアルゴリズムで一手を決定する手法である。

合議の考え方を思考ゲームの手の選択に用いた研究としては、1985年頃から行われたAlthoferらによる3-Hirnがある³⁾。3-Hirnは2つのチェスプログラムが別々の思考により出力した候補手を、十分に強い人間のプレイヤーが選択するという合議手法である。Althoferらは、この手法で合議を行うことで、元のプログラムよりもElo Ratingにして200程度強くなることを示した。彼らはさらに囲碁や他の思考ゲームでも同様の実験を行って、その有効性を示している。

これらの結果は、複数のコンピュータが挙げる手の中に単体で選んだときよりもよい手が含まれているという可能性を示唆している。すなわち、挙げられた手の中から適切な方法で一手を選択することができれば、より強くなる可能性を示唆している。

3-Hirnの研究結果を受けて、2003年に柴原らは2-Hirnという手法を提案している⁴⁾。これは3-Hirnから人間による選択を除き、コンピュータのみで指し手を決定する手法である。特定の条件下で効果があることが示されたが、現在までのところ汎用的な手法としては確立されていない。

本研究の元になる研究として、2009年3月に、埴らは合議アルゴリズムの基礎となる実験結果を5五将棋を題材に

して報告した⁵⁾。この実験では、4種類のプレイヤーを、5五将棋プログラム「千分ノ壱里眼」を变形することによって作成した⁶⁾。探索の末端で駒の取り合いを計算する静止探索の有・無と、評価関数のパラメタを独自に調整したものと5五将棋大会優勝プログラムのK55⁷⁾が公開していたデータを参考にして変更したものの組み合わせである。指し手は単純多数決により選択し、意見が2:2等に分かれた時はオリジナルの千分ノ一里眼(静止探索有り・独自評価関数)の手が選択される。4つのプレイヤー単体を相手にした100局ずつの対戦結果を表1に転載する。100局という少ない対局数のため厳密な有意差検定は行えないが、すべての単体に対して合議側が勝ち越している。この実験から、多数決による合議が有効である可能性が示唆された。

ここで行った合議アルゴリズムは非常に単純であり、プロセッサ間的高速な通信を必要としない。複数の思考プログラムを全く独立に実行し、各プログラムが指した手の中から実際に指す手を選択する。

表1 5五将棋における合議実験

プレイヤーはA: 静止探索有り・独自評価関数、B: 静止探索有り・評価関数変更、C: 静止探索無し・独自評価関数、D: 静止探索無し・評価関数変更である。

対戦相手	A	B	C	D
合議側勝率	61%	71%	58%	79%

$$\text{勝率} = \text{勝数} / (\text{勝数} + \text{敗数})$$

次章では、合議によりMinimax探索を行うシステムを構築する一般的な手法を提案する。これは、乱数を用いて複数のゲームプログラムを生成する方法であり、最低一つのプログラムさえ準備できれば合議システムの構成が可能となる。

この手法を用いた「文殊」は今年5月に開催された世界コンピュータ将棋選手権において3位の好成績を収めた⁸⁾。これは、コンピュータ将棋としては疎結合計算環境の活用で成功した初めての事例である。

指し手を選択する方法は色々考えられるが、本稿では単純多数決を用いた。単純多数決とは、最も多くのプログラムによって支持された手を選択する方法である。

3章以降では、この単純多数決による合議を様々な形で適用して、単体のプログラムより強くなるのかを調査し、そのメカニズムについて考察する。

2. 乱数による複数プレイヤーの生成

合議システムを設計するにあたり問題となる点として、強い思考プログラムを複数用意しなければならないということがある。本稿では、単一の思考プログラムを用いて容易に多数のプレイヤーを作り合議させる手法を提案する。その方法は、プログラムの評価関数によって算出される評価値にある大きさの乱数を加えるというものである。個々のプレイヤーに異なる乱数系列を与えることによって、それぞれ異なる形勢判断を持つプレイヤーを生み出すことができる。ここでは、乱数を加えることによりどの程度プログラムが弱くなるのか、また、どの程度の大きさの乱数を加えると多様な意見を得ることができるのか、といった疑問を検証する。

評価関数に与える乱数は、正規分布 $N(0, D^2)$ に従って生成した。なお、与える乱数値は局面のハッシュキーに対し

て割り当てた。すなわち、同じ乱数系列を持つプレイヤーが同じ局面に異なる評価値をつけることはない。

表2に、後述の合議 Bonanza 対 Bonanza の実験において、乱数を加えた16個の Bonanza が各局面で挙げた異なる指し手の数を示す。正規乱数の標準偏差 D が大きくなるほど意見が分かれていることが確認できる。 $D=25$ (歩の交換値の1/8程度)でも50%の確率で票が割れている。

表3は、評価値に乱数を与えることによる棋力への影響を調査したデータである。標準偏差 D の正規乱数を加えた Bonanza と通常の Bonanza を探索量別に1000局ずつ対戦させ、勝率を測ったものである。 D は Bonanza における歩の交換値=202を基準として $1/1(202)$, $1/2(101)$, $1/4(50)$, $1/8(25)$ の4通り調べた。探索量は探索深さで制限しており、両対局者とも同じ深さの探索で指し手を決定する。表の勝率は、乱数を加えた方の Bonanza から見たものになっている。

表2 静的評価関数に正規乱数 $N(0, D^2)$ を加えた Bonanza の指し手の分布

候補手数	1	2	3	4	5	6	7	8	9	10	11	12	13	Total
D=25	18521	9393	4913	2203	813	255	55	16	6	0	0	0	0	36175
D=50	15300	9837	5783	2843	1208	386	124	21	11	0	0	0	0	35513
D=101	12360	9500	6542	3792	1883	713	244	70	17	5	2	0	0	35128
D=202	9156	8387	6829	4932	3018	1564	621	222	66	18	7	2	1	34823

数値は該当局面の数

表3 Bonanza の評価値に乱数を与える影響

深さ / D	25	50	101	202
1	48.22%	43.93%	30.08%	9.11%
2	49.55%	47.45%	38.20%	19.64%
3	51.41%	46.42%	37.34%	21.64%
4	49.90%	50.00%	46.85%	34.47%
5	48.48%	49.50%	45.68%	33.20%
6	51.52%	49.65%	46.98%	37.97%

勝率 = 勝数 / (勝数 + 敗数)

データから、D=25 程度の乱数では有意な弱化はみられず、D=50 でも十分な探索量の下では通常の Bonanza と同程度の強さであるとわかる。D=101 以上になると明確に勝率が下がっているが、探索量の増加とともに弱化した分を取り戻している傾向がみられる。これは評価関数の精度の低下を、探索が補っているためと考えられる。

3. Bonanza の自己対戦による合議実験

合議の効果を検証する第一段階として、Bonanza を用いて自己対戦で実験を行った。先に述べた乱数を用いた方法で複数のプレイヤーを用意し、Bonanza 合議 対 通常の Bonanza の勝率を調べた。先に述べた通り合議の方法は単純多数決で、最も多くのプレイヤーが支持した手を採用する。ただし、最多意見が同数の票で複数ある場合は、その中からランダムに一つを選択することとした。

実験は1個のプレイヤーあたり1手20万ノードの探索の場合と、1手40万ノードの探索の場合の2通り行った。合議側のプレイヤー数は1, 4, 8, 16とした。プレイヤー数1というのは合議無しで単に乱数を加えた Bonanza である。Dは25, 50, 101, 202の4通り調べた。なお、Bonanza が序

盤の定跡データベースを使用している間、合議は行わない。各条件1000局ずつの実験結果を表4.表5.に示す。表のデータは、合議側から見たものである。

実験の結果、適切な大きさの乱数の下では合議側がほとんどの場合有意に勝ち越すデータが得られた。すなわち、乱数を用いた合議は効果があったといえる。また、プレイヤー数を多くしてもさほど効果が上がらないことが示唆された。なお参考として、二項分布を用いた仮説検定によれば、1000局中、有意水準5%では527勝、有意水準1%では537勝以上で有意に強いと言える。

4. YSS による合議実験

Bonanza 合議 対 Bonanza の実験では合議が有効であることが示唆されたが、この実験は自己対戦であるため、自分自身にのみ強くなっていた可能性を否定できない。また、Bonanza 以外でも有効であるかどうか不明である。

将棋プログラム YSS の開発者である山下氏は、先述の乱数を用いた合議実験を、YSS 合議 対 Bonanza で追試した。また、山下氏から YSS のプログラムをご提供頂き、我々の手で YSS の合議実験を追試した。これらの実験について紹介・報告する。

表4 自己対戦による合議実験(一手20万ノード)

M / D	25	50	101	202
1	49.50%	48.03%	43.72%	33.03%
4	54.27%	51.76%	54.43%	42.07%
8	53.13%	54.49%	53.37%	46.04%
16	53.48%	57.65%	52.48%	50.65%

勝率 = 勝数 / (勝数 + 敗数)

表5 自己対戦による合議実験(一手40万ノード)

M / D	25	50	101	202
1	47.83%	48.89%	45.98%	35.08%
4	55.56%	53.24%	52.21%	45.43%
8	55.19%	57.42%	54.59%	48.89%
16	53.07%	55.99%	55.70%	50.20%

勝率 = 勝数 / (勝数 + 敗数)

表 6 筆者らによる YSS の合議実験

M / D	0	6	12	25	50	100	200
1	46.6%**	45.6%	47.8%	48.2%	41.9%	41.4%	41.0%
3	—	50.0%	48.7%	52.8%	50.9%	51.6%	52.5%
8	—	51.4%	52.5%	53.2%	55.4%	50.5%	53.5%

勝率 = 勝数 / (勝数 + 敗数)

**： 2000 局の勝率

実験は、乱数を用いた多数決合議をそのまま YSS に適用した YSS 合議と Bonanza との対戦で、思考量は YSS 1 プレイヤ当たり 40 万ノード, Bonanza 10 万ノードである。

我々の実験の結果を表 6 に示す。対局数は基本的には 1000 ずつだが、**の印をつけたデータは比較の基準となるため、2000 局とした。表中の勝率は、YSS 側からみたものである。なお、YSS において歩の交換値は 200 であり、乱数の標準偏差はこれを基準に決定している。

M=1, D=0 はつまり、合議でない通常の YSS である。このときの勝率を基準に合議をした場合のデータを見ると、いずれも勝率が向上していることがわかる。合議は YSS においても効果があり、また自己対戦でなくても有効であると言える。山下氏による実験でも、同様の結果が出ている。

Bonanza の実験では歩の交換値相当の標準偏差の乱数を与えたとき、合議をしても勝率がオリジナルを上回ることはなかったが、YSS においては歩の交換値相当でむしろ高いパフォーマンスを示すという結果が得られた。YSS の

方が乱数の影響を受けにくい性質を持つ可能性がある。

5. 探索深さ別合議実験

乱数を用いた合議でなぜ強さが向上するのか、いくつかの推測はあるが理論的なことはまだわかっていない。これを解明する第一歩として、合議の効果が探索の深さと関係しているかを調べた。

実験の方法は、先の探索ノード数指定の実験とほぼ同じである。異なるのは、探索量をノード数ではなく深さで制限している点である。ただし、Bonanza は探索の末端で静止探索を行ったり、王手等によって探索深さを延長したりするため、深さ n が n 手先までのゲーム木全域という意味ではない点に注意されたい。あくまで、探索量を表す指標の一つである。

プレイヤ数 16 の場合についてデータを取り、表 3. で示したプレイヤ数 1 の場合と比較した。次の表 7 に実験結果を示す。

表 7 探索深さ別合議実験(M=16)

深さ / D	25	50	101	202
1	51.87%	50.05%	48.88%	35.44%
2	51.35%	51.10%	47.54%	35.44%
3	51.71%	51.46%	51.52%	36.38%
4	54.54%	53.71%	55.21%	44.23%
5	51.82%	54.46%	53.15%	51.51%
6	53.44%	55.87%	58.63%	53.23%

勝率 = 勝数 / (勝数 + 敗数)

結果をみると、合議をする場合は探索量が増えるほど、同じ探索量の相手に対して強くなる傾向がある。つまり、合議による強さの向上が、探索量が多いほど大きいといえる。しかし、表3に示した合議をしない場合にも探索量の増加にともなって勝率が向上しているため、単体の強さの向上がその理由と考えることもできる。この実験結果だけでは、合議そのものの効果が探索と関係があるかどうかには結論を出すことはできない。

しかし、この手法においては探索量が増えるほど効果を発揮する可能性が高いということがわかった。

6.3種の将棋プログラムによる合議実験

「独立に思考する別々の思考アルゴリズムの将棋プログラムを組み合わせる強くなるのか？」ということが本研究のそもそもの興味である。特に、実験用に準備したプログラムではなく、既存の最強プログラムにおける有効性が最強アルゴリズムを作るという目的からすれば、実用化の要といえる。本報告では、幸運にもYSSの開発者である山下氏、GPS将棋の開発チームであるTeam GPSの協力が得られ、Bonanzaと合わせて3種のプログラムによる合議実験が実現した⁹⁻¹¹⁾。これらのプログラムは、いずれも世界コンピュータ将棋選手権での優勝経験を持つ強豪である。

結果を表8に示す。合議プレイヤーは、これら3種のプログラムの単純多数決により作成した。意見が1:1:1に分かれたときは、Bonanzaの手を採用した。思考量は1秒当たりの探索ノード数を目安に制限し、Bonanza 10万ノード、YSS 40万ノード、GPS 15万ノードとした。合議対Bonanza、合議対YSS、合議対GPSのそれぞれについて1000局ずつ対局させた。なお、勝率は1000回の対戦から引き分けを除いて求めた。それぞれのプログラムに付属する序盤データベースを使用、棋譜の内容に偏りは見られなかった。

表8 Bonanza, YSS, GPS 将棋を用いた合議実験。

対戦相手	Bonanza 4.1.2	YSS	GPS
合議側勝率	64.26%	73.65%	72.24%
Bonanza	50.00%*	70.31%	59.10%

勝率 = 勝数 / (勝数 + 敗数)

*理論値

表8を見ると、合議が単体のプログラム全てに対してかなり高い勝率を収めていることがわかる。

この原因として、合議をすることにより、序盤データに含まれる低頻度の悪手が選ばれにくくなるという理由が考えられる。この序盤データ合議の公平性を検証するために、探索量を一手10万ノードにして、序盤データ利用時のみ3人で合議するBonanzaと通常Bonanzaの勝率を求めてみたところ、51.16%であった。この結果、有意な勝率の向上は認められず、少なくともBonanzaのみの合議に関しては、序盤の定跡データによる合議の不公平ではないことは示された。

このように高い勝率を収めた原因については、現在調査中であるが、異種プログラムによる多数決合議でも何らかの効果が期待できることが示唆される結果であると考えている。

7.まとめ

既存の将棋プログラムの組み合わせからなる合議システムを設計し、その性能を検証した。このシステムは疎結合並列処理を念頭に置いて設計されたものであり、各プログラムとの通信データ量は局面情報と思考結果(指し手)のみである。また、指し手を決定のアルゴリズムには最も単純な多数決を採用した。

合議アルゴリズムには3つの主な利点がある。1つ目は、先に述べたように疎結合な計算機環境を利用することができる点である。これにより、これまででない規模の計算資源を利用することが可能になることが期待される。2つ目は、非常に単純で汎用性が高い点である。この手法がどのような思考プログラムで有効であるかは調査中だが、実

装可能性でいえば、あらゆる思考プログラムに適用することができる。また、本報告で挙げた方法で示すように、たった一つの思考プログラムでも容易に合議を行うことが可能である。3つ目は、プログラム改良のこれまでにない方向性である。これまで提案されてきたコンピュータ将棋

のアルゴリズムはほぼ全て、“探索の量”の改善か“探索の質”の改善かの二つに分類できる。しかしここで示す提案手法は、そのどちらにも該当しない。全く新しい改良の方向を示していると言える。

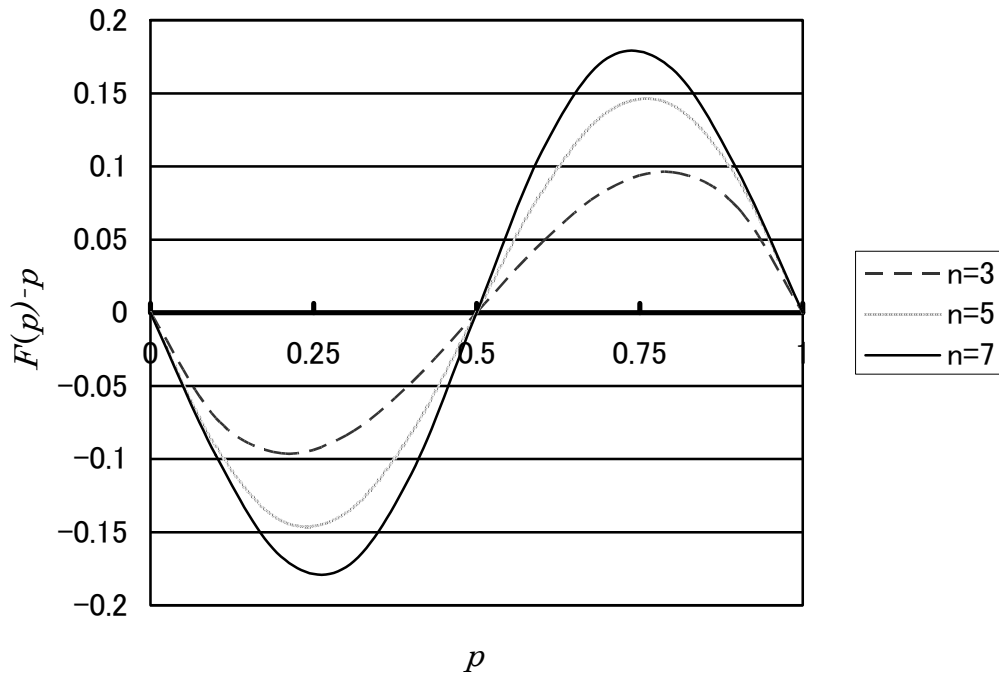


図1 多数決と単体の正解率の差 $F(p) - p$

単純多数決による合議が有効である理由の考察として、将棋の指し手ではなく正解/不正解の2値の選択肢のみがある単純な問題に置き換えて考えてみる。このときグループが正解を得るのは過半数が正解を選ぶときなので、 n 個の個体の個々の正解率を p とすると、グループの正解確率 $F(p)$ は次の式のように表せる。

$$\begin{aligned}
 F(p) = & p^n \\
 & + {}_n C_{n-1} p^{n-1} (1-p) \\
 & + {}_n C_{n-2} p^{n-2} (1-p)^2 \\
 & + \dots \\
 & + {}_n C_{\frac{n+1}{2}} p^{\frac{n+1}{2}} (1-p)^{\frac{n+1}{2}}
 \end{aligned}$$

(n は奇数とする)

$F(p)$ の値が p を上回れば、多数決の効果があるといえる。図1は、 p と $F(p) - p$ の関係を表したグラフである。なお n が偶数の場合、半数が正解を選び、半数が不正解を選ぶときの正解率が0.5であると仮定すると、 $F(p) - p$ は $n-1$ のグラフと一致する。

このグラフからは、単体の正解率 p が0.5を超えると、多数決によって正解率が向上するということがわかる。また、その向上幅はプレイヤー数が多いほど大きくなる。将棋の指し手の選択は2値の問題ではないので、この考察がそのまま将棋で適用できるとは考えていないが、多数決による合議の有効性に関する何らかの説明になっているかも知れない。

また、今回報告した実験では、合議側と対戦相手の使用する計算資源が公平でない。これは従来の並列アルゴリズム

ムでは疎結合並列処理を効率よく行うことができず、同等な計算資源を想定することができないためである。合議方法に改善の余地があることも含めて、計算資源の利用効率を議論することは現時点では難しい。しかし、従来の密結合並列処理したプログラムに対して、本手法を組み込むことは容易であり、計算資源効率を比較するのではなく、組みあわせることが出来る新しい並列化の方向性を示した点で本手法は優れていると言える。

今回の報告では、合議システムの構成法として2つの手法を提案した。一つのプログラムから複数のプログラムを生成し合議する手法と、強いプログラムを複数用意して単純合議する手法である。いずれの手法においても、トップレベルにある既存プログラムの勝率を上昇させることが示された。

本研究はまだ発展段階にあり、合議における指し手決定法に改善の余地がある。また、各プレイヤーの指し手の不一致度等から局面の難しさを見積もる等の応用も考えられる。

謝辞

本研究の実験を遂行するにあたって、YSSの開発者である山下氏にはYSSの実験に関する情報や、YSSのプログラムを提供して頂いた。また、GPS将棋の開発チームであるTeam GPSにはGPS将棋を提供して頂き、特に金子氏にはGPS将棋のプログラム改変へのサポートや、実験遂行に関して多くの助言を頂いた。この場を借りて深く謝意を表したい。

本研究は社団法人情報処理学会から、共同研究による助成を受けている。

参考文献

- 1) "Cramming more components onto integrated circuits", Electronics Magazine 19 April 1965
- 2) R. Hyatt, B. Suter, and H. Nelson, "A Parallel Alpha/Beta Tree Searching Algorithm," Parallel Computing 10 (1989) 299 - 308
- 3) Althofer, I. and Snotzke, R. G. : Playing Games with Multiple Choice System, Computer and Games, pp.142 - 153 (2002).
- 4) "2-Hirn", 柴原 一友, 後藤智章, 乾信雄, 小谷善行, 第8回 ゲーム・プログラミング ワークショップ 2003 pp.59 - 66 (2003)
- 5) 思考アルゴリズムにおける最適合議システム, 埴雅織, 伊藤毅志, 第3回エンターテインメントと認知科学シンポジウム pp.72 - 75 (2009)
- 6) 千分ノ壱里眼は著者(小幡)が作成した5五将棋プログラム。2008年UEC杯5五将棋大会COM部門3位。
- 7) 柿木義一, 5五将棋における評価関数の自動学習, 第5回 E&C 研究ステーション招待講演発表資料, http://homepage2.nifty.com/kakinoki_y/free/55shogi.pdf (2008)
- 8) コンピュータ将棋の新しい波4:合議アルゴリズム「文殊」-単純多数決で勝率を上げる新技術, 伊藤毅志, 情報処理 50 (9)pp.887 - 894 (2009)
- 9) AI将棋 Version 16 (e frontier, 2008) に採用されている市販思考エンジン、山下宏作。
- 10) GPS Shogi Revision 2114, <http://gps.tanaka.ecc.u-tokyo.ac.jp/gpsshogi/>; コンピュータ将棋の新しい波:3.最近のコンピュータ将棋の技術背景とGPS将棋, 金子知適, 情報処理 50 (9) 878-886 (2009)
- 11) Bonanza は著者(保木)が作成した将棋プログラム。http://www.geocities.jp/bonanza_shogi/ にてソースファイルが公開されている。市販版も存在する(マグノリア, 2008)