

将棋における合議アルゴリズム ——多数決による手の選択

伊藤 毅 志^{†1} 小幡 拓 弥^{†2}
杉山 卓 弥^{†3} 保木 邦 仁^{†1}

本論文では、将棋プログラムの新たな並列処理手法を提案する。このアルゴリズムは、複数の思考プログラムの候補手の中から一つの手を選択する手法である。このアルゴリズムを合議アルゴリズムと呼ぶ。本論文では、将棋における合議手法の提案と評価を行い、また単一プログラムからでも乱数を用いた合議手法を提案しその有効性も示した。さらに、YSS, GPS 将棋, Bonanza 等の有名な強豪プログラムをこの合議アルゴリズムで組み合わせることで、その各々のプログラムよりも強くなることを示した。

Consultation Algorithm in Shogi — A Move Decision by Majority

TAKESHI ITO,^{†1} TAKUYA OBATA,^{†2} TAKUYA SUGIYAMA^{†3}
and KUNIHITO HOKI^{†1}

A new algorithm that runs on a computer with interconnected processors has been designed for Shogi. The algorithm adopts consultation between many individual players. We call the algorithm "Consultation Algorithm". In this paper, we proposed and evaluated the consultation approach in Shogi, and proposed how to generate multiple players from a single program by using the random number and expressed the effectiveness of this method. Applying a simple majority rule to select a move from multiple moves, the consultation algorithm improves the performance of computer Shogi programs. It is also demonstrated that the consultation algorithm consisting of three famous Shogi programs: YSS, GPS and Bonanza plays better games than any of the three programs individually.

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約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年代後半からプロセッサ単体の性能は伸び悩んでいる。そのため現在では、複数のプロセッサを連結することで性能向上を図るのが一般的である。

これを受けて近年、ソフトウェアのさら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 $\alpha\beta$ 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) および 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⁾。しかし、これらの手法は疎結合な計算環境では効率良く動作しない。 $\alpha\beta$ 法に対する疎結合並列化においては決定的なアルゴリズムがいまだになく、今後の展開が期待されている研究領域である。本研究のほかにも、将棋を題材とした分散並列実行の研究も報告されている³⁾。

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」とは、複数の独立した思考システムが個々に導いた候補手の中から、何らかの方法で一つの手を選択するアルゴリズムと定義する。

複数のプログラムの候補手から一つの手を選択するという研究に関しては、1985年頃から行われた Althöfer らによる 3-Hirn がある⁴⁾。3-Hirn は二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から、十分に強い人間のプレイヤーが一手を選択するという手法である。Althöfer らは、この手法を用いることで、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。彼らの手法は、候補手の中から手を選択しているのが「人間」であり、人間がどのように選択しているのかという方法については言及していないので、合議アルゴリズム自体には踏み込んでいない。

^{†1} 電気通信大学
University of Electro-Communications

^{†2} キヤノン株式会社
Canon Inc.

^{†3} 日本アイ・ビー・エム株式会社
IBM Japan Corporation

しかし、これらの結果は、複数のコンピュータがあげる手の中には、単体で選んだときよりも良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、複数のコンピュータがあげ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、単体よりも強くなることを意味している。

本研究では、本論文に先立って行った研究^{5),6)}の実験データを拡充し、多数決による合議アルゴリズムの有効性について、以下の手順で検証していく。

まず、乱数を用いて一つのプログラムを複数のプレイヤーにするアルゴリズムを提案する。2章では、乱数の大きさに対応して、どのように指し手がばらつくのか、また、探索の深さと棋力との関係についても調べ、乱数を与える影響に関する基礎データを示す。3章では、この基礎データをもとに、乱数による複数のプレイヤーを作り、多数決によって合議させる「乱数合議」の手法により、多数決合議の有効性を調べる。さらに、4章では、複数の異種の思考プログラム (Bonanza, YSS, GPS 将棋) による多数決合議の有効性について述べる。5章では、これらの実験結果について考察し、6章では本論文全体をまとめる。

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、ある程度強さの揃った思考プログラムを複数用意しなければならないという点がある。一般に、強さの近い複数の思考プログラムを複数用意することは困難であり、合議アルゴリズムの有効性を示す実証的研究を難しくしてきた。本章では、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。

その方法とは、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、それぞれ微妙に異なる形勢判断を持つプレイヤーを容易に複数生み出すことができる。この手法は、評価関数を持つコンピュータ将棋プログラムで広く一般的に用いることが可能であると考えている。

評価関数に加える乱数が Minimax 探索に与える影響は自明ではなく、ルートノードにおける候補手からランダムに手を選ぶこととはまったく異なる。ただし、一般的には棋力が低下する。ここでは、乱数を与えることによりプログラムがどの程度弱くなるのか、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、といった点を検証する。

表 1 に、3章で述べる合議 Bonanza 対 Bonanza の実験 (Bonanza 4.0.4, 一手 40 万ノー

表 1 静的評価関数に正規乱数 $N(0, \sigma^2)$ を加えた Bonanza の指し手の分布

Table 1 Distribution of moves of Bonanza which added the regular random number $N(0, \sigma^2)$ to the static evaluation function.

候補 手数	1	2	3	4	5	6	7	8	9	10 ~ 13	計
$\sigma=25$	18,521	9,393	4,913	2,203	813	255	55	16	6	0	36,175
$\sigma=50$	15,300	9,837	5,783	2,843	1,208	386	124	21	11	0	35,513
$\sigma=101$	12,360	9,500	6,542	3,792	1,883	713	244	70	17	7	35,128
$\sigma=202$	9,156	8,387	6,829	4,932	3,018	1,564	621	222	66	28	34,823

数値は該当局面の数

表 2 Bonanza の評価値に乱数を与えたことによる棋力への影響

Table 2 Influence on the strength by having given the random number to the evaluation value of Bonanza.

深さ/ σ	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)

ド)において、乱数を加えた 16 個の Bonanza が各局面であげた候補手の数を示す。たとえば初手 7 六歩が 10 票、2 六歩が 5 票、5 六歩が 1 票であった場合、候補手数は 3 である。これを対象の実験中に現れたすべての局面について集計した。標準偏差 σ 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。最も小さい $\sigma = 25$ (歩の交換値の 1/8 程度) でも 50% の割合で票が分かれている。

表 2 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 σ の正規乱数を加えた Bonanza と通常の Bonanza を探索量別に 1,000 局ずつ対戦させ、勝率を測った。 σ 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 からみたものになっている。

表 3 Bonanza の合議実験結果 (一手 20 万ノード)

Table 3 A result of the consultation experiment by Bonanza (200,000 node per a move).

M/σ	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%

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

表 4 Bonanza の合議実験 (一手 40 万ノード)

Table 4 A result of the consultation experiment by Bonanza (400,000 node per a move).

M/σ	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%

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

データから、 $\sigma = 25$ 程度の乱数では有意な棋力の低下はみられず、 $\sigma = 50$ 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分かる。 $\sigma = 101$ 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。

この手法により、単一の思考プログラムから擬似的に強さの近い複数の思考プログラムを作る手法を提案することができた。以降の章では、この弱化した思考プログラムを組み合わせることで、その弱さを補ってより強い思考プログラムを導く「合議アルゴリズム」が提案できるのかを検証していく。

3. 自己対戦による多数決合議アルゴリズム

3.1 自己対戦による結果

合議アルゴリズムの効果を検証する第一段階として、将棋プログラム Bonanza 4.0.4 を用いて自己対戦で実験を行った⁷⁾。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 の勝敗を調べた。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アルゴリズムを用いた。ただし、最多数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一手 20 万ノードの探索の場合と、一手 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 M は 1, 4, 8, 16 とした。プレイヤー数 1 というのは合議なしで単に乱数を加えた Bonanza である。 σ 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 1,000 局ずつの実験結果を、表 3, 表 4, および、図 1, 図 2 に示す。表の勝率は、合議側からみたものである。

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というデータが得られた。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、1,000 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いえる。なお、以下、表中の * は有意水準 5%、** は有意水準 1% で有意に強くなったことを示すこととする。

図 1 は、表 3 の結果をグラフとして表したものである。 $\sigma = 50, 202$ では、プレイヤー数が増加するに従って、勝率が高くなっていることが分かる。 $\sigma = 25, 101$ では、プレイヤー数が 4 のとき最大で、その後はプレイヤー数を増やしても勝率が伸びなくなっている。

図 2 は、表 4 の一手 40 万ノードと探索数を増やした場合の結果である。全般に一手 20 万ノードに比べて勝率が高い傾向にある。最も好成绩だったのは、 $\sigma = 50$ で、プレイヤー数が 8 のときであった。

Bonanza では、多数決合議の有効性が示されたので、同様の乱数合議の手法で、別の将棋プログラム GPS 将棋でも同様の結果が得られるかを確かめることにした。GPS 将棋は 2009 年世界コンピュータ将棋選手権で優勝した強豪プログラムの一つである^{8),9)}。

実験は Bonanza の場合と同様に、乱数を用いて複数のプレイヤーを生成し、GPS 将棋合議と GPS 将棋の勝敗を調べることで行った。探索量は一手 15 万ノードとし、 σ は歩の交換値を基準に 1/1, 1/2, 1/4, 1/8 とした。実験結果を表 5 に示す。なお、GPS 将棋においては歩の交換値は 200 と近似する。表 5 をみると、GPS 将棋でも Bonanza のときと同様に、評価値に乱数を加えると単体のプログラムは弱くなるが、合議することによって、単体よりも有意に強くなるという合議の効果を確認することができた。

3.2 YSS による他プログラムに対する合議実験

ここまでの実験では自己対戦において、合議が有効であることが示唆されたが、自己のプログラムに対してのみ強くなっているという可能性が残る。合議が他のプログラムに対し

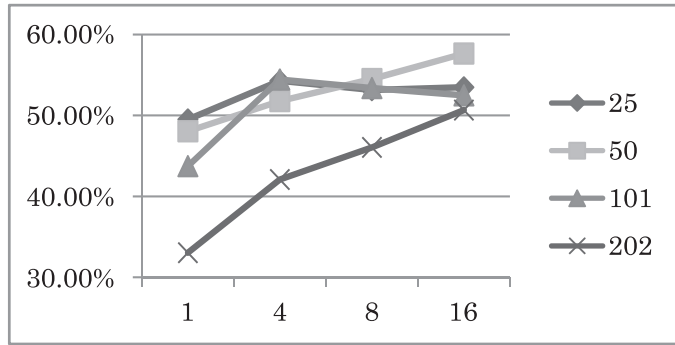


図1 Bonanzaの合議実験(一手20万ノード)

Fig. 1 A result of the consultation experiment by Bonanza (200,000 node per a move).

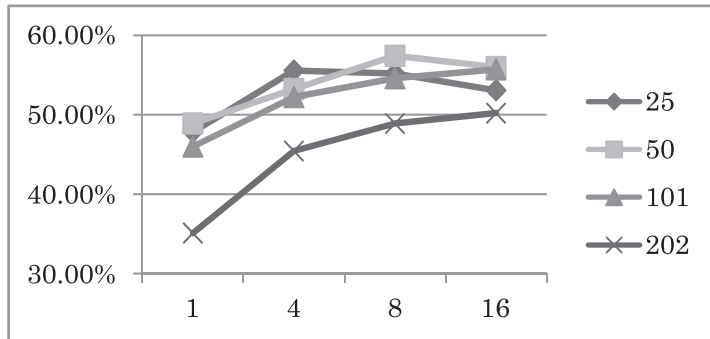


図2 Bonanzaの合議実験(一手40万ノード)

Fig. 2 A result of the consultation experiment by Bonanza (400,000 node per a move).

表5 GPS将棋の合議実験

Table 5 A result of the consultation experiment by GPS-Shogi.

M/σ	25	50	100	200
1	51.76%	46.85%	45.80%	35.77%
4	55.23%**	55.86%**	55.77%**	50.85%

勝率 = 勝数 / (勝数 + 敗数)

表6 YSSによるBonanzaに対する合議実験

Table 6 A result of the consultation experiment to Bonanza by YSS.

M/σ	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

ても有効であるかを検証するため、将棋プログラムYSSを上述と同様の方法で合議させて、Bonanzaを対戦相手とした実験を行った。YSSは、2007年世界コンピュータ将棋選手権で優勝する等、つねに上位で活躍している強豪プログラムの一つである¹⁰⁾。

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YSS1プレイヤーあたり40万ノード、Bonanza10万ノードとした。

実験の結果を表6に示す。対局数は1,000局ずつだが、††の印をつけたデータは比較の基準となるため、これに限り2,000局対戦した。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, σ=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みると、いずれも勝率が向上していることが分かる。合議はYSSにおいても効果があるばかりか、他のプログラムに対しても効果があることが示された。また、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らなかったが、YSSにおいては歩の交換値相当でも高いパフォーマンスを示すという結果が得られた。

4. 異種の将棋プログラムによる合議実験

「合議アルゴリズム」は、そもそも独立に思考する別々の思考プログラムの手の中から一つの手を選択するアルゴリズムであるので、可能であれば複数のまったく異なる思考プログラムによる合議アルゴリズムの効果を検証したい。今回は、Bonanza, GPS将棋, YSSの三つのプログラムが揃ったので、この3種類のプログラムによる多数決合議の効果について調べることにした。

本実験における合議方法は、これら3種のプログラムの単純多数決とした。ただし、意見が1:1:1に分かれたときはBonanzaの手を採用することとした。これは、以下に述べる

実験条件下で3種のプログラムを総当りで1,000局ずつ対局させたところ、Bonanzaの勝率が最も高かったためである。すなわち、GPS将棋とYSSが一致してBonanzaの手を否定しない限り、つねにBonanzaの手を選択することになる。

各プログラムの思考量は1秒あ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将棋 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1,000局ずつ対局させ、勝敗を調べた。3種のプログラムの総当りの対局結果と合わせて、結果を表7に示す。なお、序盤の定跡データベースはそれぞれのプログラムに付属するものを使用した。棋譜の重複についても調べたが、観察されなかった。

表をみると、合議が単体のプログラムすべてに対してかなり高い勝率を収めていることが分かる。対Bonanzaに対して、64.26%の勝率を上げていることから、異種の将棋プログラムによる合議の効果ははっきりと認められた。

この結果から、序盤の定跡データベースの利用に不公平がなかったかという疑問が生じる。たとえばBonanzaでは、序盤データベースからの手選択は、データベース作成に用いた棋譜中で登場した手の頻度に応じた確率に応じて選ばれる。このデータベースの手について多数決を行った場合、確率の高い手がより選ばれやすくなると考えられる。このことが序盤データ利用の不公平となり、対局結果に影響を与えている可能性がある。

この序盤データの利用の公平性を検証するために、序盤データ利用時のみ3プログラムで合議し、後は単体のBonanzaとして振る舞うBonanzaと通常Bonanzaの勝敗を求めた。序盤の定跡から外れた後は、通常のBonanza対通常のBonanzaとなる。探索量は一手10万ノードとした。結果は、合議側からみて51.16%であった。この結果からは有意な勝率の

向上は認められず、少なくともBonanzaのみの合議に関しては、序盤の定跡データ利用による不公平性は認められなかったといえる。

5. 考 察

5.1 探索深さと多数決合議の関係

乱数を用いた多数決合議でなぜ強さが向上するのかについては、いくつかの推測はあるが理論的なことはまだ分かっていない。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とり、表3で示したプレイヤ数1の場合と比較した。表8にその実験結果を示す。

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。

しかし、この結果から、多数決合議に関しては、探索量が増えるほど効果を発揮する手法である可能性が示唆された。これは、探索と合議アルゴリズムとの関連を示すものであると考えられる。

表7 Bonanza, YSS, GPS将棋を用いた合議実験

Table 7 The consultation experiment by using Bonanza, YSS and GPS-Shogi.

対戦相手⇒	Bonanza	YSS	GPS将棋
Bonanza	50.00% [†]	70.31%**	59.10%**
YSS	29.69%	50.00% [†]	63.46%**
GPS将棋	40.90%	36.54%	50.00% [†]
合議	64.26%**	73.65%**	72.24%**

勝率 = 勝数 / (勝数 + 敗数)
[†]理論値

表8 探索深さ別合議実験 (M = 16)

Table 8 The consultation effect against the search depth (M = 16).

深さ/σ	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)

5.2 計算資源と合議の効果について

合議アルゴリズムの一つの大きな特徴として、疎結合の並列化が実現できる点があげられる。このことは、従来の密結合の並列化手法とは異なる軸の研究であることを意味している。

密結合の並列環境においても、並列度が高くなるにつれて勝率の上昇に陰りがみられることが指摘されている。本手法は、さらにそのうえでの改善の可能性を示すものであると考えている。合議アルゴリズムによる並列計算に関する議論では、しばしば従来の並列化手法と比較して計算資源の効率化が図れているのかという指摘があるが、合議アルゴリズムはそれらと競争するものではなく、併用してさらなる改善を目指すことを目的としている。

合議アルゴリズムが上述のような目的であることを考慮したうえで、その資源効率について検証した実験データを表9に示す。表は、Bonanzaを用いて、一手あたりの探索ノード数が10万である場合を1倍として、多数決合議、逐次計算、資源効率が並列度の平方根である仮想的な並列手法の三つについて、計算資源の増加に対する効果を比較したものである。多数決合議においては、資源の増加は多数決に加わるプレイヤー数の増加を意味し、4プレイヤー(4倍)、8プレイヤー(8倍)、16プレイヤー(16倍)の3通り検証した。逐次計算は、一つのプレイヤーに単にN倍の量の思考をさせた場合であり、20万ノード(2倍)、40万ノード(4倍)、80万ノード(8倍)、160万ノード(16倍)の4通り検証した。従来の並列手法であるDTSでは、資源効率は並列度の平方根よりやや良いとされている。ここでは、資源効率が並列度の平方根である並列手法を想定し、逐次計算で14.1万ノード(2倍)、20万ノード(4倍)、28.2万ノード(8倍)、40万ノード(16倍)とすることで、代わりとした。

対戦相手は、一手あたりの探索量を20万ノードに固定したGPS将棋とした。

結果より、従来手法が使える環境においては、提案手法よりも従来手法の方が性能的に優れていることが示された。

表9 資源効率の比較
Table 9 Comparison of calculation resource efficiency.

資源	多数決合議	逐次計算	仮想的な並列手法
1倍		55.5%**	
2倍	—	66.1%**	64.2%**
4倍	61.2%**	71.7%**	66.1%**
8倍	63.7%**	79.1%**	71.5%**
16倍	61.5%**	81.2%**	71.7%**

勝率 = 勝数 / (勝数 + 敗数)

しかし、提案手法は、実装容易性や耐故障性において優れており、従来手法との併用も可能である。繰り返すが、提案手法は従来手法にとって代わるものではなく、併用や使い分けを前提にしたものであることを強調しておく。

5.3 多数決による意思決定の効果について

異種の将棋プログラムによる多数決合議に効果がみられたことについて、ここでは、一般の意思決定問題に置き換えて、一つの考察を試みる。すなわち、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非常に単純化した意思決定問題をn個の個体によるグループで解決する課題に置き換えて考えてみる。このとき多数決でこの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がpであったとき、グループの正解確率F(p)は次の式のように表せる。

$$F(p) = \sum_{i=0}^{(n-1)/2} n C_{n-i} p^{n-i} (1-p)^i \quad (n \text{ が奇数の場合})$$

F(p)の値が単体の正解率pを上回れば、多数決の効果があるといえる。図3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、単体の正解率pが0.5を超えるとき、多数決によって正解率が向上することがいえる。2値で選択する問題であれば、サイコロを振っても0.5の正答率は得られ

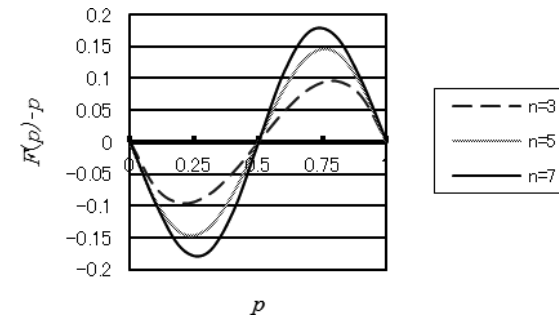


図3 多数決と単体の正解率の差: F(p) - p

Fig. 3 The difference of the correct answer rate between majority by group and a single player: F(p) - p.

るので、チャンスレベル以上の判断力を持つ判別器であれば、合議の効果が得られることが分かる。また、その効果はプレイヤー数が多いほど大きくなる。ただし、この考察において、各個体の p は互いに独立であるという前提がある。各 p に相関がある場合、その度合いが大きいほど向上幅は小さくなり、完全に一致する場合向上幅は 0 になる。本論文で報告した実験において、異なるプログラム間の合議が単一プログラムから生成されたプレイヤー群による合議よりも高いパフォーマンスを示したのは、このことと関係していると考えられる。

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、多数決による意思決定問題の有効性を示す一つの説明になっている可能性はある。

6. おわりに

本論文では、まず、単一のプログラムから複数の思考プログラムを生成する「乱数合議」の手法を提案した。そして、その複数の思考プログラムを用いて多数決によって一つの手を決める合議アルゴリズムについて述べ、その有効性を示した。具体的には、適当な大きさの乱数を与えることによって、自己対戦により合議の勝率が上昇すること、他のプログラムに対してもその効果があることを示した。さらに、三つの異なる思考アルゴリズムのプログラムを用いた多数決合議でも、その有効性を示し、自己対戦よりも効果的である可能性が示唆された。これらの結果をまとめると表 10 のようになる。

単一のプログラムから導いた複数の思考プログラムによる多数決合議で棋力が向上する原因の調査は現在遂行中であるが、ゲーム木探索との関連について示唆される結果が得られた。また、そのスケーラビリティには、一定の限界があることも示唆された。

さらに、異種の思考プログラムの合議の効果について、正解不正解を選ぶ非常に単純化した意思決定モデルを仮定したところ、一般に個々がチャンスレベル以上に知的な判断を下す

表 10 合議手法の効果のまとめ

Table 10 The conclusion of the effect for each consultation approach.

合議方法	乱数合議		複数プログラムによる 異種合議
	対:自己対戦	対:他プログラム	
効果	○	○	◎

○…効果あり ◎…高い効果あり

システムであれば、多数決により合議の効果が得られることが論理的に導かれた。将棋のような複雑な問題における多数決の合議アルゴリズムに関しても、この説明が成り立つかどうかは分からないが、一定の説明になっていると考えている。現在もそのメカニズムの解明に努めており、様々な試みを行っているところである¹¹⁾。

最後に、多数決による合議アルゴリズムの大きな利点として、以下の二点をあげておく。一つは、プロセッサ間の通信なく疎結合な計算機環境を利用することができる点である。したがって、従来の密結合の並列化とは競合せずに、それにプラスする形で計算資源を使えるというメリットがある。もう一つは、非常に単純で再現性、一般性が高い点である。実験で示したとおり、標準的な思考アルゴリズムを持つ複数のプログラムで本手法は有効であった。このことから、幅広いプログラムで本手法は有効である可能性が高い。また、本手法は Minimax 探索を行うすべてのプログラムに容易に実装可能であり、最低一つのプログラムを用意すれば実現できることも示された。この二つの点において、実用に足る有効な手法であることを示すことができた。

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宏氏には YSS のプログラムや実験に関する情報を提供していただいた。また、GPS 将棋の開発チームである Team GPS には、GPS 将棋を提供していただき、特に金子知適氏には、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いただいた。この場を借りて深く謝意を表したい。なお、本研究は、一般社団法人情報処理学会の共同研究による助成を受けている。

参 考 文 献

- 1) Moore, G.E.: Cramming more components onto integrated circuits, *Electronics*, Vol.38, No.8 (Apr. 1965).
- 2) Hyatt, R.M., Suter, B.E. and Nelson, H.L.: A Parallel Alpha/Beta Tree Searching Algorithm, *Parallel Computing*, No.10, pp.299–308 (1989).
- 3) 金子知適, 田中哲朗: 最善手の予測に基づくゲーム木探索の分散並列実行, 第 15 回ゲームプログラミングワークショップ 2010, 箱根 (Nov. 2010).
- 4) Althöfer, I. and Snatzke, R.G.: Playing Games with Multiple Choice System, *Computer and Games*, pp.142–153 (2002).
- 5) 埴 雅織, 伊藤毅志: 思考アルゴリズムにおける最適合議システム, 第 3 回エンターテイメントと認知科学シンポジウム, pp.72–75 (2009).
- 6) 小幡拓弥, 杉山卓弥, 保木邦仁, 伊藤毅志: 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る強いプレイヤーを作れるのか?, 第 14 回ゲームプログラミング

ワークショップ 2009, pp.51-58 (2009).

- 7) Bonanza は保木邦仁が作成した将棋プログラム, 入手先(http://www.geocities.jp/bonanza_shogi/), 市販版も存在する(マグノリア, 2008).
- 8) GPS Shogi は GPS 将棋チームが開発した将棋プログラム. Revision 2114, 入手先(<http://gps.tanaka.ecc.u-tokyo.ac.jp/gpsshogi/>).
- 9) 金子知適: 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, 情報処理, Vol.50, No.9, pp.878-886 (2009).
- 10) YSS (商品名: AI 将棋) は山下宏が開発した将棋プログラム. ここでは, Version 16 (e frontier, 2008) に採用されている市販思考エンジンを使用.
- 11) 小幡拓弥, 保木邦仁, 伊藤毅志: 乱数合議の有効性に関する一考察, 第 15 回ゲームプログラミングワークショップ 2010, pp.9-14 (2010).

(平成 23 年 4 月 4 日受付)

(平成 23 年 9 月 12 日採録)



伊藤 毅志 (正会員)

1994 年名古屋大学大学院工学研究科情報工学専攻修了. 工学博士. 同年電気通信大学情報工学科助手. 2007 年より, 同助教. 2010 年より, 電気通信大学情報理工学研究科助教. 人間の思考過程, 学習過程に関する研究に従事. 現在は特に思考ゲーム等を題材にした認知科学的研究に興味を持つ. 著書に『先を読む頭脳』(新潮社, 共著)ほか. 日本認知科学会, 人工知能学会等会員. コンピュータ将棋協会, コンピュータ囲碁フォーラム各理事.



小幡 拓弥 (正会員)

2009 年電気通信大学情報工学科卒業. 2011 年電気通信大学大学院情報工学専攻博士前期課程修了. 同年キヤノン株式会社入社, 在職中. 情報処理学会山下記念研究賞受賞.



杉山 卓弥

1986 年生まれ. 2008 年慶應義塾大学工学部情報工学科卒業. 同年慶應義塾大学大学院理工学研究科開放科学専攻コンピュータ科学専修前期博士課程に入学し, 2010 年に修了. 同年日本アイ・ピー・エム株式会社に入社. 在職中. 将棋を題材とした人工知能的研究に興味を持つ. 第 14 回ゲームプログラミングワークショップ 2009 研究奨励賞受賞.



保木 邦仁 (正会員)

2003 年東北大学大学院理学研究科化学専攻修了(理学博士). 同年トロン大学化学科博士研究員. 2006 年より, 東北大学にて研究員, 助手として教育研究に従事. 2010 年より, 電気通信大学特任助教. 著書に『ボナンザ VS 勝負脳—最強将棋ソフトは人間を超えるか』(角川新書, 共著).