

異種プログラム間における楽観合議について

竹内 聖悟^{1,a)}

概要: 将棋や囲碁において、多数決などの複数プレイヤーを使った合議による指手決定が成功を収めている。合議についての研究では、評価値が最も高い手を選択する「楽観」合議が多数決よりも良い結果を得ている。また、囲碁においてプレイヤーの多様性が高いチームが強いプレイヤー一つから作られるチームよりも強くなるという研究がある。本研究では、これらの研究に基づき「異種楽観合議」を提案する、すなわち異なるプレイヤー間で楽観合議を行う手法である。異なるプレイヤー間では評価関数のスケールが異なるため、単純な大小比較ではうまくいかないと予想される。本研究では、評価値から勝率を予測する関数を作成し、その関数を使った楽観合議、勝率楽観合議を提案する。将棋プログラムを用いて実験を行い、いずれの合議も単体よりも性能が良くなることを確認した。その後、単純楽観合議と勝率楽観合議による対戦実験を行い、性能について多数決合議と同等以上となることを確認した。

Optimistic Consultations on a Heterogeneous System

SHOGO TAKEUCHI^{1,a)}

Abstract: Move selection by a consultation system using multiple programs, such as majority voting system, has been successful in shogi and game of go. There is a study that optimistic consultation that selects the move with the highest evaluation value performed better than majority voting. Besides, there is research that systems with high diversity become stronger than systems made from single strong program.

We propose optimistic consultation on a heterogeneous system based on these studies, that is, a method to conduct optimistic consultations among different programs. On the other hand, it is expected that simple optimistic consultation will perform poorly, since the scale of the evaluation function differs among different programs. In this research, we propose a winning percentage based optimistic consultation that selects the move with the highest winning percentage calculated from the evaluation value.

We conducted tournament experiments using shogi programs and confirmed that the performance of any consultation system improves over single program. Moreover, we confirmed that the performance of the proposed method is equal to or greater than that of majority voting system.

1. はじめに

強いコンピュータプレイヤーの作成には、局面評価と探索とが重要である。将棋や囲碁、チェスのような二人零和有限確定完全情報ゲームでは alpha-beta 探索やモンテカルロ木探索が使われてきた。計算機資源の効率的な利用方法として、並列化や分散並列化の研究が行われている [8]。計算機資源利用方法の一つとして、複数のプログラムを用いた合議システムと呼ばれる手法が研究されてきた [2]。

合議システムでは多数決による手法が提案され、プログ

ラム単体よりも強くなる結果が得られた。その後、最も評価値が高い指手を選ぶ楽観合議の方が良いという実験結果が得られた [6]。また、合議を構成するプログラムの多様性について、強いプログラム単一のチームを形成するよりも、多様性のあるチームによる多数決合議の方が強くなる結果が得られている [1]。本研究ではこれらに着目し、楽観合議を多様性のあるチームで行うことを提案する、従来の手法よりも性能改善が期待される。

一方、異なるプログラムを使った楽観合議を考えると、評価値のスケールが異なると単純な比較では適切な楽観が行えないことが考えられる。例えば、将棋プログラムにおいて歩を 100 点と評価するプログラムと 128 点と評価する

¹ 高知工科大学 情報学群

^{a)} takeuchi.shogo@kochi-tech.ac.jp

プログラムを考えると、自身が優勢のときは、歩が128点のプログラムの方が選ばれやすくなる。歩に限らず評価項目は多岐に渡るため、単純な比較やスケールリングは難しい。他、共通でない評価項目が存在する場合の扱いも自明ではない。

異なるプログラム間で評価を共通的に扱うために、評価値に何らかの補正を加えることが必要となる。探索の枝刈りや延長において、勝率と評価値の関係を利用した研究がある。互角な局面における100点差と大勢が決した局面における100点差とが同等かという問題意識のもと、評価値から得られる勝率から得られる情報量に基づいて枝刈りや延長の判断を行う手法が、竹内らによって提案されている [9]。また、評価関数の学習において勝率を利用する手法が研究されており、他に、Satoらは比較学習の際に、棋譜の勝敗と評価値との関係を加味して学習を行った [3]。この手法では、シグモイド関数を使い、評価値から勝率を計算している。評価値に対する勝率を、棋譜の勝敗から計算、それを図示して特徴の評価に用いる evaluation curve と呼ばれる手法が Takeuchi らによって提案されている [7]。このように、プログラムの評価値を勝率へと変換することで、異なるプログラム間でも共通の指標として取り扱うことができると考えられる。

本研究では、合議システムの性能改善を目的として、異なるプログラムを使った楽観合議、異種楽観合議を提案する。プログラム間で評価値スケールが異なる場合に問題があると考えられ、解決のため評価値から勝率を予測する関数を作成し、それを利用することを提案する。以降、スケールを考えずに行う異種楽観合議を「異種単純楽観合議」、勝率を利用する楽観合議を「異種勝率楽観合議」と呼ぶ。

2. 関連研究

チームを形成して指手を選択する手法として多数決合議 [2]、楽観合議 [6] が知られている。また、チームに参加するプログラムの多様性についての研究があり、これらの研究を説明する。

提案手法では評価値と勝率の関係を利用する他、実験の性能評価においてもその関係を利用する。評価値と勝率の関係を表す evaluation curve [7] とそれによる性能評価について説明する。

2.1 多数決合議

Obata らは、チーム内の多数決によって指手を選択する手法を提案した [2]。この際に、プログラムの評価値について乱数を加えることで、同種であるが少し異なるプログラムを作成した。実験結果では、多数決システムは単体に有意に勝ち越す結果を得ている。さらに、異なる複数のプログラムによる多数決も実験している。

以下、同じプログラムによるチーム、又は同じプログラ

ムに乱数を加えるなどして形成するチームを「同種」と呼び、異なる複数のプログラムから形成するチームを「異種」と呼ぶ。

2.2 楽観合議

Sugiyama らは Obata らと同様に同種のプログラムに乱数を加えたチームを対象として、最も評価値の高い指手を選ぶ「楽観」合議 [6] を提案した。単純に指手の情報だけを使うのではなく、評価値の情報も利用することで改善できることを示した。Obata らの多数決合議と同様に同種のプログラムに乱数を加え、少しずつ異なるプログラムを用意し、それらの中で評価値が最も高い指手を採用するというシステムである。実験では、楽観合議は多数決合議よりも良い結果を得た。

2.3 合議に対する多様性の影響

Obata らの実験において、同種のプログラムだけを使うのではなく、異なるプログラムを使った多数決合議も試されており、同種のプログラムを用いるよりも良い結果が得られている [2]。

また、Marcolino らは強いプレイヤーによる合議よりも多少弱くても多様性のあるチームによる合議の方が強くなることを囲碁プログラムを題材として示している [1]。実験では、4つの異なるプログラムと、その中で最も強いプログラム Fuego のバリエーション2つの計6プログラムを用いた。計算機資源を4または6利用する際に、Fuego だけでチームを作る場合、並列探索を行った場合、異なるプログラムでチームを作る場合とで対戦実験を行った。Fuego はモンテカルロ木探索を行うため、同じプログラムを複数用意してもそれぞれ異なる結果を出すことがある。また、チームでの指手決定は多数決合議を行っている。結果として、4つの場合は多様性のあるチームが Fuego だけのチームよりも有意に良い結果を得るなど、多様性の高さが重要であることが示唆される結果が得られている。

2.4 評価値と勝率

局面評価値は局面における形勢判断を数値化したものであり、勝率を何らかの形で近似したものとも言える。局面評価値の学習において、目的関数として棋譜の勝敗と評価値から得られる勝率との差が用いられる。例えば、Alpha Go [4] の Value network の学習や Alpha Go Zero [5] の学習、2017年に世界コンピュータ将棋選手権で優勝した将棋プログラム elmo では学習に勝率を用いる手法^{*1}が使われており、その後多くの将棋プログラムが同様の手法を用いるようになった。他にも、Sato らは比較学習の際に、棋譜の勝敗と評価値との関係を加味して学習を行った [3]。

^{*1} http://www2.computer-shogi.org/wcsc27/appeal/elmo/elmo_wcsc27_appeal_r2_0.txt 5月25日閲覧

評価値と勝率の関係を図示するものとして, evaluation curve[7]がある. まず, 局面に対して, 棋譜の勝敗とプログラムによる評価値のペアをつける. 同じ評価値の局面を集め, その勝敗の合計から勝率を計算するという手順を行う. 実際には, 評価値そのものではなく範囲を使う.

評価関数の特徴について重みが適切であるか, またはある特徴が有効であるかを調べるのに用いられた. ある特徴がオンになっている局面集合とオフになっている局面集合とを集め, それぞれで evaluation curve をプロットする. この時, evaluation curves が分かれていると, 評価値の大小関係と勝率の大小関係に逆転が生じ, 勝率の低い手を選ぶ可能性があるため, このような特徴は調整の余地があるというアイデアであった.

評価値に対して標準化などを行った場合, 異なるプログラム間であっても, 同じスコアであるならば同じ勝率となる必要がある. シグモイド関数による評価値から勝率への変換については, 元のデータを十分近似できているのであれば, 自然と達成できる.

3. 提案手法

楽観合議が多数決合議よりも良い結果を得ており [6], 多様性の高い方が良い結果を得ていることから [1], 多様性のある, 異種プログラム間での楽観合議を提案する. しかし, 楽観合議では評価値の高い指手を選ぶが, これは評価関数のスケールがほぼ同一のプログラム間で成立する手法である. あるプログラムの評価値が他よりも少し高めであるとすると, そのプログラムの指手が選ばれやすくなり, 本来の楽観合議の効果は得られないと考えられる. 異なるプログラム間で評価値のスケールを考えずに楽観合議を行う手法を, 「異種単純楽観合議」または「単純楽観」と呼ぶ.

問題は, プログラム間で評価値のスケールが異なることに起因しており, 評価値のスケールを揃えることで改善が期待される. そこで, 局面の評価値と勝率の関係を考え, 評価値 x から勝率 y を予測するようなシグモイド関数 $\frac{1}{1+\exp(-w_1x+w_2)}$ を使う. この w_1, w_2 については, Takeuchiらが提案した evaluation curve[7]の調整法などを用いて, プログラムごとに計算する. 本研究では, このシグモイド関数を使い, 異なるプログラム間の評価値を共通の指標へと変換し, それによる楽観合議を提案する. この手法を「異種勝率楽観合議」または「勝率楽観」と呼ぶ.

evaluation curve では, 異なる局面集合においても同じカーブが得られることが, 正しい評価の条件となっていた. 調整後の指標についても同様のことが期待される. すなわち, 同じスコアに対しては同じ勝率を持つことが期待される. これが満たされないと指し手選択時に, 勝率の低い指し手を選ぶ可能性が生じるためである.

表 1 プログラム単体同士の対局結果

	Apery	YaneuraOu	技巧	nozomi
A	-	0.698 (690 16 294)	0.6895 (685 9 306)	0.772 (760 24 216)
Y	0.302 (294 16 690)	-	0.525 (497 56 447)	0.538 (507 62 431)
G	0.3105 (306 9 685)	0.475 (447 56 497)	-	0.5695 (561 17 422)
N	0.228 (216 24 760)	0.462 (431 62 507)	0.4305 (422 17 561)	-

4. 実験

提案手法の有効性を示すために, 将棋を題材として実験を行った. まず, 実験に用いるプログラム同士の対戦を行い, その強さを計測した. その結果に応じて, 合議に用いるプログラムを選び, 多数決合議や単純楽観合議や勝率楽観合議による対戦を行い, その性能を計測した.

4.1 実験環境

実験は, Intel Xeon CPU E5-2630 v4 @ 2.20GHz 10core 2CPU, メモリ 96GB, Ubuntu 16.04 の環境で行った.

対戦プログラムとして Apery^{*2}, YaneuraOu^{*3}, 技巧^{*4}, nozomi^{*5} を用いた. これらは 2018 年 4 月頃のプログラムを利用している.

各プログラムが用いる評価関数の特徴について述べる. Apery では 3 駒関係と手番, YaneuraOu にはゼロ評価関数 epoch 8^{*6} を使用したため 3 駒の関係 (手番なし), 技巧は 2 駒関係, nozomi は 3 駒関係と手番となっている. 後の実験結果から, Apery 以外の 3 プログラムでチームを構成した, 駒の関係だけを見てもそれぞれに違いがあり, 多様性の高いチームになっていると考えられる.

実験時には 1 プログラムあたり 1core を使用し, 1 手 5 秒の探索を行う. これはプログラム単体でも合議でも同様である. 対戦は, 棋譜の 30 手目の局面から開始する. 全 500 局面を用意し, それぞれ先後を入れ替えて対戦を行うため, 全体で 1,000 局の対戦が行われる.

4.2 予備実験: 単体の強さ計測

まず, プログラム単体の強さを計測するため対戦を行った. 対戦結果を表 1 にまとめる.

実験から, Apery の性能が最も高く, 以下は YaneuraOu, 技巧, nozomi の順と考えられる. 以降の実験では, Apery 以外の 3 プログラムでチームを作り, Apery との対戦から

*2 <https://github.com/HiraokaTakuya/apery> 5月25日閲覧

*3 <https://github.com/yaneurao/YaneuraOu> 5月25日閲覧

*4 <https://github.com/gikou-official/Gikou> 5月25日閲覧

*5 <https://github.com/saihyou/nozomi> 5月25日閲覧

*6 <https://drive.google.com/open?id=0Bzbi5rbfN85NMhd00EUxcUVJQW8>, 5月25日閲覧

性能の改善を見る。

4.2.1 シグモイド関数の調整

評価値から勝率を予測するシグモイド関数 $f(x) = \frac{1}{1 + \exp(-w_1 x + w_2)}$ の重み w_1, w_2 を調整した。evaluation curve[7] の手法にならない、シグモイド関数から得られる勝率と棋譜の勝敗との誤差を最小化する最小二乗法 (LSE) , および勝率と棋譜の勝敗との尤度を最大化する最尤法 (MLE) によって調整を行った。棋譜のある局面に対して、評価値をつけ、それから得られる勝率と棋譜の勝敗との差を最小化する。

棋譜の局面に対して評価値をつける必要があり、計算コストがかかる。今回の実験では、そのコストを避けるため、プログラム単体で行った対戦棋譜につけられた評価値を用いてこの調整を行った。YaneuraOu, 技巧, nozomi による合議を行い、Apery との対戦を行うため、対 Apery 戦の棋譜を除いた。また、1 棋譜から多くの局面を取り出すことを避けるため、60 手ごとにランダムに 1 手取り出すこととした。YaneuraOu の場合、対技巧, nozomi 戦の棋譜、計 2,000 局を使い、30 手目以降の局面から、60 手の間でランダムに 1 手を選び、評価値とシグモイド関数から得られる勝率と棋譜の勝敗との誤差を最小化することで調整を行った。evaluation curve では、評価関数の値または静止探索の値であったが、今回は 1 手 5 秒の探索を行った評価値を利用している。

調整の結果を図 1, 表 2 にまとめる。nozomi が選ばれにくい傾向となった。また、YaneuraOu の evaluation curve と、最小二乗法 (LSE), 最尤法 (MLE) で調整したシグモイド関数を図 2 に示す。技巧および nozomi については付録の図 A-1, A-2 を参照されたい。どのプログラムでも最小二乗法によるシグモイド関数の方が evaluation curve によく合致したため、以降の勝率楽観合議では、最小二乗法によって得られたシグモイド関数を利用した。なお、本実験の evaluation curve では評価値を 200 点ごとに区切り、その中の勝敗から勝率を近似している。この際、各区切りは 50 以上のデータがある箇所だけをプロットした。

図 4 に evaluation curve を描いた際の、各区切りにおける局面数を示す。発生頻度について各プログラムでは大差がないことがわかる。その一方、全プログラムの evaluation curve をプロットした図 3 から、nozomi は評価値 0 の時の勝率が他よりも著しく低いことがわかり、単体プログラム同士の勝敗の影響が生じていることがわかる。これから、nozomi の手が楽観としては選ばれにくくなると予測される。

これらの局面集合における評価値の平均と分散を表 3 にまとめる。評価値の分布に正規分布を仮定し、平均 μ と分散 σ から評価値 x の標準化を行うと、標準化された評価値は $\frac{x-\mu}{\sigma}$ となる。このスコアと勝率の関係、evaluation curve を図 5 に示す。異なるプログラム間では同じスコア

表 2 プログラムごとのシグモイド関数の係数

プログラム	LSE		MLE		局面数
	w_1	w_2	w_1	w_2	
YaneuraOu	0.002895	0.1613	0.001082	0.1349	3,835
技巧	0.002600	0.1638	0.001659	0.1260	3,775
nozomi	0.002484	0.5750	0.001061	0.4533	3,696

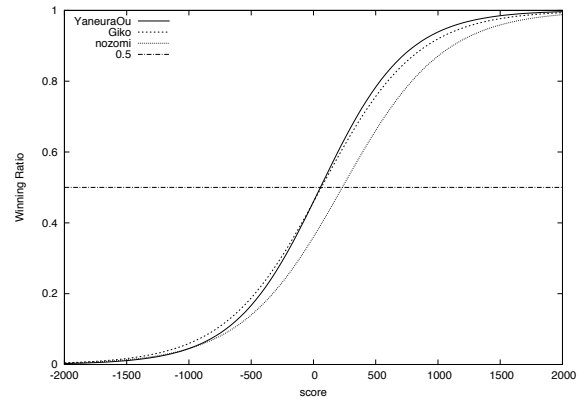


図 1 各プログラムのシグモイド関数

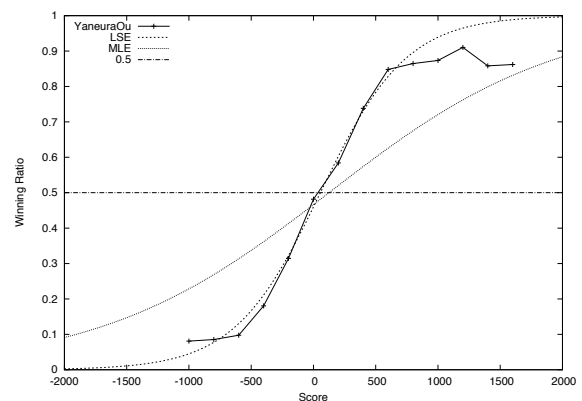


図 2 YaneuraOu の対戦棋譜から得た evaluation curve と最小二乗法, 最尤法によるシグモイド関数. 実線が evaluation curve.

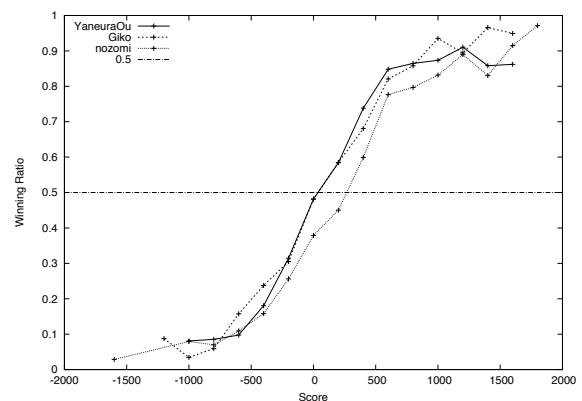


図 3 各プログラムの evaluation curve

でも勝率が大きく異なるケースが散見され、平均と分散による標準化は今回の問題に適していないことがわかる。例えば、3 プログラムのスコアがすべて 0.0 の場合、勝率が最も高い技巧の指し手が選ばれるべきであるが、このスコアではそれが反映されておらず、選ばれるとは限らない。

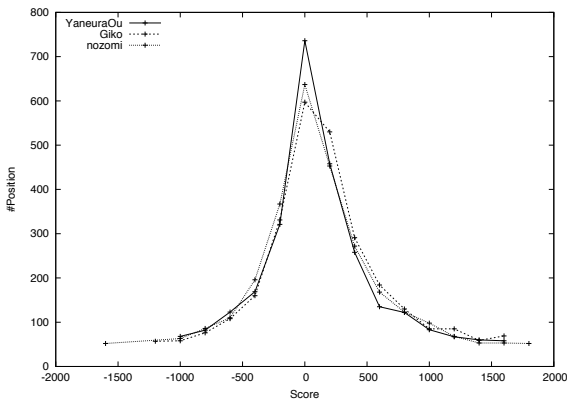


図 4 評価値とその頻度, 全プログラムについて

表 3 プログラムごとの評価値の平均と分散

プログラム	平均	分散	局面数
YaneuraOu	99.09	2228.1	3,835
技巧	334.00	1925.9	3,696
nozomi	420.16	2107.4	3,775

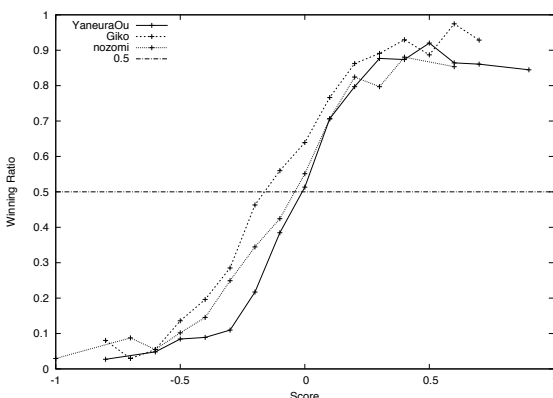


図 5 標準化した評価値による evaluation curve

4.3 合議システムの性能

Apery 以外の 3 プログラムによるチームを形成し、そのチームでの多数決合議や楽観合議を行い、対戦による性能計測を行った。多数決合議と単純楽観合議、勝率楽観合議とを Apery と対戦させた。結果を表 4 にまとめる。これに加え、各プログラム単体の対 Apery 戦の勝率を図 6 に図示する。なお、多数決合議において多数派がない時はリーダーとした YaneuraOu の指し手を採用することとした。

合議に参加するプログラム単体では、勝率が 2 割から 3 割強であったことから、合議は全般的に性能が改善していることが分かる。また、各合議システムと単体プログラムの勝率に有意な差がないかカイ二乗検定したところ、いずれも有意水準 5% で有意な差が得られた。一方、各合議システム同士の勝率についてカイ二乗検定を行ったところ、勝率楽観合議と多数決合議との間には有意水準 5% で有意な差があったが、他の組み合わせについては有意な差は得られなかった。

表 4 合議と Apery の対戦結果

	対 Apery
勝率楽観合議	0.427 (423 8 569)
単純楽観合議	0.412 (409 6 585)
多数決合議	0.379 (362 34 604)

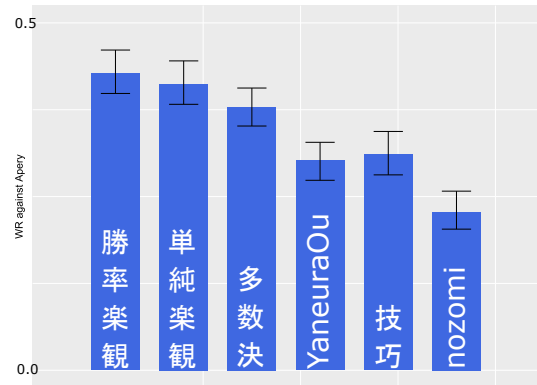


図 6 各手法の対 Apery 勝率と信頼区間

表 5 合議同士の対戦結果

	単純楽観合議	多数決合議
勝率楽観合議	0.507 (492 30 478)	0.499 (473 52 475)
単純楽観合議	-	0.498 (477 42 481)

4.4 合議同士の対戦実験

前節の Apery との対局に続き、性能評価のため合議システム同士の対戦を行った。対戦結果を表 5 に示す。前節の Apery 戦の勝率では、勝率楽観合議と多数決合議との間に有意に差があったが、今回の対局結果ではどの組み合わせにおいても有意な結果は得られなかった。

4.5 指し手採用率など

各合議システムの振る舞いの差を調べるため、指し手採用率を調べた。対 Apery 戦の棋譜からデータを取った結果を表 6、図 7 にまとめた。なお、楽観合議では評価値が最大でなくても選ばれた手と一致するなら採用率に含めているため、各システムの採用率の総和は 1.0 を超える。

多数決は常に多数派が選ばれるためもあり、他よりも全般的に数値が高い。さらに多数決合議においては、全員の指し手が異なる場合は YaneuraOu の指し手が選ばれることが影響してか、YaneuraOu の指し手採用率が高くなっている。勝率楽観では、nozomi の採用率が低く、単純楽観でも勝率楽観でも技巧の採用率が最も高いなど、合議システムごとに違いが現れた。

各システムでの投票の様子を調べたところ、どのシステムでもおおよそ変わらず、全員合意が 0.47、YaneuraOu と技巧が同じ手の場合が 0.11、YaneuraOu と nozomi が 0.14、技巧と nozomi が 0.14、各 1 票となったのは 0.14 程

表 6 各プログラムの指手の採用率。データは対 Apery 戦の棋譜から得た。楽観においては評価値が最大でなくても、選ばれた手と一致した場合には採用されたとみなしている。

	YaneuraOu	技巧	nozomi
勝率楽観合議	0.7503	0.8080	0.6549
単純楽観合議	0.7189	0.7626	0.7209
多数決合議	0.8571	0.7184	0.7416

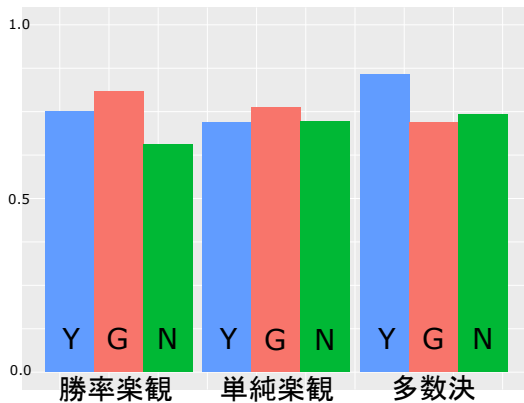


図 7 各合議システムごとの指し手採用率。(Y:YaneuraOu, G:技巧, N:nozomi)

度であった。当然ながら、表 6 の多数決合議と一致する。

5. 議論

勝率楽観合議は多数決合議に対し、対 Apery 戦の勝率では有意な差を得た一方で、直接対戦させた場合は有意な差が得られなかった。まずは、提案手法の性能がある程度改善した理由を考察する。

勝率楽観合議が用いるシグモイド関数の効能について考えると、シグモイド関数から nozomi の指手が選ばれにくくなることが考えられる。このことは図 7 から確認できる。そこで、nozomi を除き、YaneuraOu、技巧の 2 プログラムによる単純楽観合議と Apery との対戦を行った。勝率楽観合議と同等の結果が得られるとすると、シグモイドを利用する効果は nozomi の手を採用しにくくなったことだけで、楽観合議には寄与していないこととなる。結果は、楽観合議から見て勝率 0.4035 (399 勝 9 分 592 敗) となり、勝率楽観合議ほどの改善は見られておらず、単に nozomi が除去されることだけが性能改善の原因でないことがわかった。

単純な楽観合議でもある程度強くなったことについては、歩の点数が統一されていることから、あまり大きな違いが生じていないことが原因として考えられる。また、十分に強いプログラムであるので、ある程度スケールしていることが考えられる。

勝率楽観合議と多数決合議との対戦では有意な差が得られなかった。理由としては、いずれの合議も構成プログラムが同じであることが考えられる。

最尤法による結果が最小二乗法に比べて、evaluation

curve にフィットしていない結果が見られた。evaluation curve の調整に使う棋譜が足りていないことが考えられる。また、調整に使う棋譜が対局から得られたもので、その後の対戦実験では同じ開始局面を利用しているの、偏りが生じていることが考えられる。他にも、自分の評価値から自分の対局結果を予測しているため予測精度は高いようにも思えるが、対戦相手に依存する要素もあるため、正負のどちらの効果を持つのかは不明である。再現性の問題もあり、あまりプログラムに依存しない棋譜や勝敗を利用した方が良いだろうと思われる。

6. おわりに

楽観合議の有用性と、多様性の重要性に関する研究を背景に、異種プログラム間での楽観合議を提案した。異なるプログラムでは評価値のスケールが異なるため、単純にはうまくいかないことが予想されたため、スケールを揃えて楽観合議を行う手法を提案した。具体的には、棋譜の勝敗と評価値の関係から勝率予測関数を作成し、それによって得られる勝率を用いた楽観合議を行うことを提案した。スケールを揃えない楽観合議を「単純楽観合議」、勝率を用いる楽観合議を「勝率楽観合議」と名付けた。将棋プログラムを用いた対戦実験を行ったところ、多数決合議も含め、単体プログラムの勝率に対して有意な差を得ることができ、合議の有用性を確認できた。また、Apery との対戦実験からは勝率楽観合議が多数決合議よりも有意に良い結果を得たが、直接の対戦実験などからは有意な差は得られなかった。単純楽観合議は対 Apery 戦において多数決合議に対して有意な差は得られておらず、勝率導入の有効性を示すことができたと考えられる。

今後の課題としては、シグモイド関数を改善するために、対戦に用いる局面と重複のないような棋譜を選ぶこと、例えば、floodgate の棋譜からレーティングがある程度以上のプレイヤー同士の対局を取り出すなどが考えられる。floodgate の棋譜を使うことで局面数を増やすことも可能となり、改善が期待できる。また、現状は 3 プログラムによる合議を行っているがこれらに加えて、第 28 回世界コンピュータ将棋選手権に参加した、新しい評価関数^{*7}を用いたプログラムを利用することで、さらに多様性を高めることが期待できる。他にも、評価関数を用いないようなモンテカルロ木探索プログラムへの適用なども考えられる。

謝辞 本研究は JSPS 科研費 17K12807 の助成を受けたものです。

付 録

^{*7} https://www.apply.computer-shogi.org/wcsc28/appeal/the_end_of_genesis_T.N.K.evolution_turbo_type_D/nmue.pdf 5 月 25 日閲覧

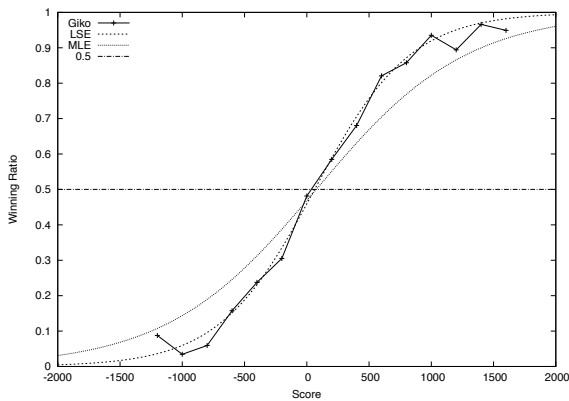


図 A.1 技巧の対戦棋譜から得た evaluation curve と最小二乗法、最尤法によるシグモイド関数。実線が evaluation curve。

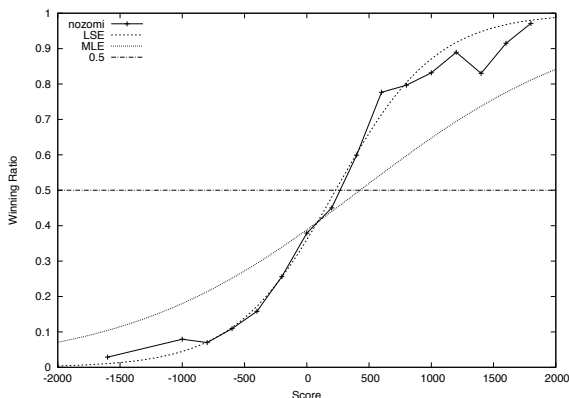


図 A.2 nozomi の対戦棋譜から得た evaluation curve と最小二乗法、最尤法によるシグモイド関数。実線が evaluation curve。

A.1 evaluation curve とシグモイド関数

技巧および nozomi について、evaluation curve と、調整されたシグモイド関数を図 A.1, A.2 に示す。

参考文献

- [1] Leandro Soriano Marcolino, Albert Xin Jiang, and Milind Tambe. Multi-agent team formation: Diversity beats strength? In *IJCAI 2013, Proceedings of the 23rd International Joint Conference on Artificial Intelligence, Beijing, China, August 3-9, 2013*, pp. 279–285, 2013.
- [2] Takuya Obata, Takuya Sugiyama, Kunihito Hoki, and Takeshi Ito. Consultation algorithm for computer shogi: Move decisions by majority. In *International Conference on Computers and Games*, pp. 156–165. Springer, 2011.
- [3] Yoshikuni Sato, Makoto Miwa, Shogo Takeuchi, and Daisuke Takahashi. Optimizing objective function parameters for strength in computer game-playing. In *Twenty-Seventh AAAI Conference on Artificial Intelligence (AAAI-2013)*, pp. 869–875, 2013.
- [4] David Silver, Aja Huang, Chris J. Maddison, Arthur Guez, Laurent Sifre, George van den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, Sander Dieleman, Dominik Grewe, John Nham, Nal Kalchbrenner, Ilya Sutskever, Timothy Lillicrap, Madeleine Leach, Koray Kavukcuoglu, Thore Grae-

- pel, and Demis Hassabis. Mastering the game of go with deep neural networks and tree search. *Nature*, Vol. 529, No. 7587, pp. 484–489, 01 2016.
- [5] David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, Yutian Chen, Timothy Lillicrap, Fan Hui, Laurent Sifre, George van den Driessche, Thore Graepel, and Demis Hassabis. Mastering the game of go without human knowledge. *Nature*, Vol. 550, pp. 354–, October 2017.
- [6] Takuya Sugiyama, Takuya Obata, Kunihito Hoki, and Takeshi Ito. Optimistic selection rule better than majority voting system. In H.Jaap Herik, Hiroyuki Iida, and Aske Plaat, editors, *Computers and Games*, Vol. 6515 of *Lecture Notes in Computer Science*, pp. 166–175. Springer Berlin Heidelberg, 2011.
- [7] Shogo Takeuchi, Tomoyuki Kaneko, Kazunori Yamaguchi, and Satoru Kawai. Visualization and adjustment of evaluation functions based on evaluation values and win probability. In *Proceedings of the Twenty-Second National Conference on Artificial Intelligence (AAAI-2007)*, pp. 858–863, 2007.
- [8] Shu Yokoyama, Tomoyuki Kaneko, and Tetsuro Tanaka. Parameter-free tree style pipeline in asynchronous parallel game-tree search. In Aske Plaat, Jaap van den Herik, and Walter Kosters, editors, *Advances in Computer Games*, pp. 210–222, Cham, 2015. Springer International Publishing.
- [9] 竹内聖悟, 金子知適, 山口和紀. 情報量に基づく探索制御手法 -チェスにおける singular extension への応用-. 第 12 回ゲームプログラミング ワークショップ, pp. 52–59, 2007.