

合議アルゴリズムにおけるリーダー決定法の検討

Leader Determining Method in Consultation Algorithm

當間 啓介*1
Keisuke TOUMA

遠藤 聡志*2
Satoshi ENDO

當間 愛晃*2
Nariaki TOUMA

赤嶺 有平*2
Yuhei AKAMINE

山田 孝治*2
Koji YAMADA

*1琉球大学大学院理工学研究科情報工学専攻

The Graduate School of Engineering and Science, University of the Ryukyus

*2琉球大学工学部情報工学科

The Department of Information Engineering, University of the Ryukyus

The validity of the consultation algorithm is shown in computer shogi. The consultation algorithm determines a move by votes of computer shogi players. Some research suggests the validity of the leader player in consultation. But, a discussion about the leader player in consultation is not enough yet. In this research, We discuss how to determine the leader whose opinion is adopted preferentially in consultation. We use shogi player's evaluation scores and the numbers of adoption in consultation to determine the leader. Experimental results suggests the validity of our proposed method.

1. はじめに

近年、コンピュータ将棋は目覚ましい発展を遂げており、女流プロとの対局に勝利するシステムも登場している。この女流プロに勝利したシステムで採用されたアルゴリズムが、合議アルゴリズムである [1]。合議アルゴリズムとは、複数思考ルーチンが出した解を基に、より良い解を選び出すアルゴリズムである。この「より良い解を選び出す」というプロセスが合議であり、合議の方法は、単純多数決合議、楽観的合議などが存在する。文献 [1] では、合議において意見が割れた際、主張が優先して採用されるリーダーが存在することによって、勝率が高くなる可能性があるという示唆がある。本研究では、適切な合議リーダーを決定する方法を検討・提案する。

2. 先行研究

2.1 5五将棋での実験

小幡らによる、本将棋よりも盤面の小さな5五将棋というゲームでの合議実験がある [1]。この実験では4つの異なる思考プログラムを合議させる。その中から一つリーダーを決めておき、意見が割れたときは、そのリーダーの意見を採用する。また、リーダーを決めないランダム決定での実験もなされている。4つのプログラムをP0～P3としたとき、相互の勝率の平均は、表1となった。なお、対局は各100局で行っている。

	P0	P1	P2	P3
勝率	63.8%	43.3%	56.8%	36.3%

表1: 5五将棋4プログラムの勝率

表1より、4プログラムは、P0 > P2 > P1 > P3の順で勝率が高い。これら強さの異なるプログラムの合議と、元の4プログラムとの対局が行われた。そのときの合議側の勝率は、表2のようになった。

	勝率
P0 リーダー	67.3%
P1 リーダー	59.0%
P2 リーダー	64.3%
P3 リーダー	56.5%
リーダー無 (ランダム)	65.3%

表2: 5五将棋合議リーダーごとの合議側勝率

表2より、4プログラム中最も強いP0を合議リーダーとしたとき、最も勝率が高いことがわかる。リーダーを決めなくても勝率は高いようだが、小幡らは、強いものをリーダーとするほうが強くなる可能性があるという示唆している。

2.2 「文殊」における合議アルゴリズム

合議アルゴリズムは、本将棋においては、「文殊」という将棋プログラムで初めて採用された [2]。「文殊」では、ソースコードが公開されている将棋プログラム「Bonanza」を使用している。Bonanzaの持つ盤面評価関数に、指定した標準偏差の乱数を加えることで、擬似的に複数の異なるプレーヤを生み出し、合議を行う。合議の方法は、最も多くのプレーヤが支持する手を最善手とする、単純多数決が採用されている。この最善手を決める投票の際、票が割れるなどして、複数の最善手が出てしまう場合は、それらの中からランダムに手を決める。合議を行うことにどれくらいの効果があるのか調べた伊藤による実験がある [2]。この実験は、乱数を加えたBonanzaを使って合議を行い、元のBonanzaと1000回対局させるといったものである。そのときの合議側の勝率は、表3のようになった。

ここで、個体数Mは、合議に参加するプレーヤの数を示し、標準偏差σは、評価関数に与える乱数の標準偏差を示している。また、勝率は、勝数 ÷ (勝数 + 敗数) というように計算されている。標準偏差σは、歩の交換値202点を基準に、その二分の一、四分の一、...というように設定して、実験している。σは、もとの評価関数が出力する評価値をどれくらいばらつかせるか、ということを決めるパラメータである。表3より、σが大きくなるほど、勝率が下がっているのがわかる。一

連絡先: 當間 啓介, 琉球大学大学院理工学研究科情報工学専攻, keisuke.touma@eva.ie.u-ryukyu.ac.jp

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%

表 3: Bonanza に対する勝率

方、個体数 M については、 M が大きくなると勝率が上がっていくという傾向がみられる。勝率が上がらない場合もあるようだが、適切なパラメータを設定してやれば、合議によって勝率を上げることができると伊藤は結論している。

3. 提案手法

先行研究での合議 [2] (以下では乱数合議と呼ぶ) では、どのプレーヤを優先させるかといったことが考慮されておらず、合議での意見割れに対しては、ランダムに選択するという方法で対応していた。乱数合議では参加するプレーヤが皆同じ思考ルーチンを使用しているため、どのプレーヤが一番優れているかという検証は行われなかったが、それでも、細かな評価値の違いは確実に存在している。また、その細かな違いの中でも、最も優れたプレーヤをリーダーとするとどうなるのか、という疑問がある。

また、序盤と終盤では戦法が異なってくるように、局面ごとに盤面評価の方法を変える必要がある。そこで、試合展開に対応して、良い評価を出しやすいプレーヤをリーダーにすることができれば、よりよい合議が可能になると考えた。

以上を踏まえ、試合展開に対応して最も優れたプレーヤをリーダーとする合議を提案する。この方法では、まず、どのようにプレーヤを評価するかを考える必要がある。プレーヤの評価基準には様々なものがあると考えられるが、本研究では、意見の採用実績と、プレーヤの返す局面評価値に着目する。

3.1 意見の採用実績に基づくプレーヤ評価法

意見の採用実績に基づいて票の重み付けをするには、そのプレーヤの意見が採用された回数を保存しておく必要がある。プレーヤ票の重み付けは、合議が満場一致 (すべてのプレーヤが同一の手を選択する場合) である限り、すべてのプレーヤに対して行われる。採用実績に差がつくのは、合議が複数意見に割れたときである。このときに重み付けが行われないプレーヤを決定する。つまり、この方法で積極的に考慮するのは、参加プレーヤの内で意見の重みを軽くすべきプレーヤである。意見の採用実績に基づくプレーヤ票重み付けのアルゴリズムを表 4 に示す。

まず、採用手へ投票された票に注目する。採用された手に投票したプレーヤがいれば、そのプレーヤの採用回数をひとつ増やす。採用されなかった手に投票したプレーヤの採用回数は増やさない。また、採用手への投票数も保持しておき、票の重み付けを行うプレーヤ数を決定するのに使用する。

5つのプレーヤで合議を行う場合を例として説明する。

例えば、採用手への投票数が5で満場一致の場合、すべてのプレーヤの票に重みが加算される。そして、投票数が4のときは、採用回数が上位4つのプレーヤにのみ重みが加算される。このように投票数が総プレーヤ数を下回る場合、採用実績の上位何件までに重みをつけるかということ、投票数に応じて決定する。

```
foreach(採用手への投票)
  プレーヤ. 採用回数 ++
  投票数 ++
end foreach
```

採用回数の降順にプレーヤをソート

```
for(i=0; i < 投票数; i++)
  プレーヤ [i]. 票 ++
end for
```

表 4: 採用実績に基づく重み付けアルゴリズム

3.2 局面評価値に基づくプレーヤ評価法

局面評価値に基づくプレーヤの評価方法では、各プレーヤの返す局面評価値を、対局開始の時点から毎回記録しておく必要がある。今回はもっとも単純な方法として、対局開始から現在までプレーヤが返した局面評価値の合計を採用した。採用実績に基づく方法では、複数のプレーヤに票が加算される場合があった。局面評価値を利用すると、合議の投票状況を考慮する必要がなくなり、合計評価値の高さによってのみプレーヤの優劣を判断することができる。プレーヤ票の重み付けは、局面評価値の合計がもっとも高いプレーヤにのみ行う。表 5 に、局面評価値に基づくプレーヤ票重み付けのアルゴリズムを示す。

```
foreach(すべての投票)
  if(自分が先手)
    プレーヤ. 合計評価値 += 票. 評価値
  elseif(自分が後手)
    プレーヤ. 合計評価値 -= 票. 評価値
  end if
end foreach
```

合計評価値の降順にプレーヤをソート

```
if(評価関数を使った探索が行われている)
  プレーヤ [0]. 票 ++
end if
```

表 5: 局面評価値に基づく重み付けアルゴリズム

局面評価値は、先手が後手かで符号が正になるか負になるかが決まる。したがって、後手の場合は、合計評価値の符号を反転させて保存しておく必要がある。

また、序盤の評価関数を使用しない場面 (Bonanza が序盤定跡データを使って手を決定している場合) では、プレーヤが局面評価値を返さないため、その段階ではプレーヤ票の重み付けは行わない。プレーヤが評価関数を使用して探索を始めた時点で、プレーヤ票の重み付けが開始される。この際、重み付けが行われるのは、局面評価値の合計がもっとも高いプレーヤとなる。

4. 実験

4.1 実験方法

実験は、合議プログラムと元プログラム (Bonanza 単体) を 1000 回連続で対局させて行った。実験には、ブレードサーバー

上の仮想マシン 10 台を使用した。

将棋プログラム同士の対局には、shogi-server [4] というネット対局用サーバーを使用した。この shogi-server を介して、単体 Bonanza と合議プログラムとの対局を行った。将棋プログラムは、Bonanza 6.0 (2012 年 2 月時点最新版) [3] を使用した。対局に使用したのは、合議側が 7 台、合議を行わない単体が 1 台となる。また、すべての Bonanza に、探索するノード数を 10 万とする制限を加えた。これは、よりフェアな条件で対局を行うための制限である。合議サーバには、乱数を加えた Bonanza を 7 台接続した。これらの Bonanza に与えられる乱数の標準偏差は 15 とした。また、それぞれに与える乱数シード値は異なる。

上記のプログラムすべてを、仮想マシン上で動作させて実験を行い、対局ログから勝率を算出し、評価した。

4.2 単純多数決合議における参加プレーヤの局面評価値推移

単純多数決合議において、参加プレーヤの返す局面評価値がどのように推移するのか調べるために実験を行った。実験方法は本論文の 4.1 項に示すような、単純多数決合議プログラム (Bonanza7 台参加) と単体 Bonanza を対局させるものとなる。

この実験で行った対局中の局面評価値の推移図を図 1 に示す。それぞれの線は、参加プレーヤの返した局面評価値を表している。また、この対局では合議側が勝利している。

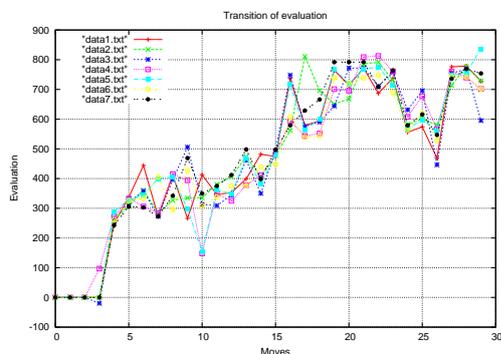


図 1: 合議参加プレーヤの局面評価値推移図

図 1 より、それぞれの参加プレーヤの返す局面評価値は、ほぼ同じような推移の傾向を示していることがわかる。しかし、全く同一の評価を返すというわけでもなく、局面ごとで、高い評価値を返すプレーヤが変化していることも見て取れる。

4.2.1 投票内容

合議のなかで、どのように投票が分かれているのか、9 手目と 10 手目の投票のログを見て調べてみる。

合議ログのそれぞれの行の項目は、

票の重み, 支持手, nps(nodes per second), 探索時間, プレーヤ名, 支持手の評価値

というような順番で並んでいる。

9 手目の合議ログを以下に示す。

```
sum = 7
1.00 2233GI nps= 270.7K 0.1s b1 266
1.00 2233GI nps= 283.5K 0.2s b2 335
1.00 2233GI nps= 239.2K 0.3s b3 506
```

```
1.00 2233GI nps= 229.8K 0.4s b4 394
1.00 2233GI nps= 261.9K 0.2s b5 298
1.00 2233GI nps= 248.8K 0.4s b6 427
1.00 2233GI nps= 260.4K 0.3s b7 469
```

9 手目の合議ログを見ると、すべてのプレーヤが 3 三銀という手を支持しており、投票は満場一致となっていることがわかる。しかし、すべてのプレーヤが同じ手を支持しているにもかかわらず、それぞれの返した評価値は異なっている。

次に、10 手目の合議ログを以下に示す。

```
sum = 3
1.00 4344FU nps= 236.7K 0.3s b2 335
1.00 4344FU nps= 316.0K 0.2s b6 308
1.00 4344FU nps= 235.1K 0.3s b7 350
sum = 2
1.00 3222OU nps= 229.1K 0.4s b1 413
1.00 3222OU nps= 234.0K 0.3s b3 313
sum = 1
1.00 5354FU nps= 242.6K 0.3s b4 148
sum = 1
1.00 6151KI nps= 226.0K 0.4s b5 154
```

10 手目の合議ログを見ると、4 四歩が 3 票、2 二玉が 2 票、5 四歩が 1 票、5 一金が 1 票、というように投票が分かれている。単純多数決なので、ここでは、最も得票数の多い 4 四歩が採用手として選ばれた。

それぞれの返した評価値について、最も高い評価値の手は 2 二玉だが、この手への投票数は 2 票であり、単純多数決によって却下される。単純多数決では、最も高い評価値を返したプレーヤの意見が採用されないという場合がある。

4.3 提案手法の評価

まず、単純多数決合議 (先行研究の合議法) と単体 Bonanza の対局実験結果を表 6 の最上段に示す。勝率は、勝数 ÷ (勝数 + 敗数) というように計算している。なお、ここに示す数値はすべて合議側についてのものである。また、表 6 には、採用実績に基づくプレーヤ票の重み付けを行った合議 (以下、採用実績合議と呼ぶ) と、局面評価値に基づくプレーヤ票の重み付けを行った合議 (以下、局面評価値合議と呼ぶ) の対局実験結果も示す。

単純多数決合議の結果を基準にして、提案手法の効果を評価する。

合議方式	勝数	敗数	引き分け数	勝率
単純多数決	555	436	9	56.0%
採用実績	571	420	9	57.62%
局面評価値	548	439	13	55.52%

表 6: 対局実験の結果

二項分布を用いた仮説検定によると、1000 局中、有意に強いといえる勝数は、有意水準 5% で 527 勝、1% で 537 勝以上である。表 6 より、提案手法の勝数はどちらとも 537 勝を超えている。提案手法は、合議アルゴリズムの持つ有効性を十分保持している。

また、単純多数決合議と採用実績合議の結果を比較すると、採用実績合議の勝数、勝率が、単純多数決合議のものよりも若干上昇しているのがわかる。つまり、対局中にプレイヤーの票重みを変化させることで、単純多数決よりも勝率が上がる可能性があるといえる。

4.3.1 採用実績合議での票重み推移

図2に、対局中に各プレイヤーの票重みが推移する様子を示す。なお、この対局では、合議側が勝利している。図中の線は、b1が乱数シード値10のプレイヤー、b2が20、b3が30... という意味である。

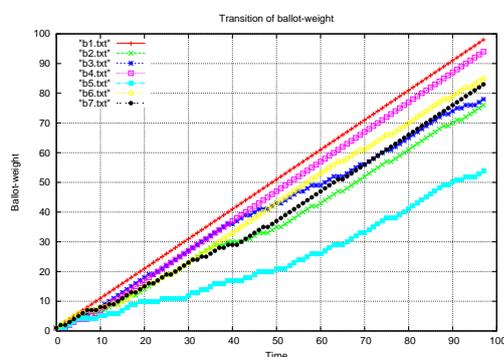


図2: 採用実績合議での票重みの推移

投票の際、票が割れなければ、基本的に重みはどんどん加算されていくので、直線的な右肩上がりのグラフとなっている。重みの1番高いメインリーダーといえるプレイヤーは、終局まで変化の様子がみられない。この理由は、票重みを加算するプレイヤーの決定法にあると考えられる。今回の実装では、票割れが生じて採用手への投票数が減った場合、採用実績の低いプレイヤーには重みの加算が行われなくなっている。つまり、採用実績が高いほど重みの加算が行われやすく、メインリーダーとなるプレイヤーは変化しづらい。

しかし、重みが2位以下のプレイヤーに関しては、順位が変動している様子が見て取れる。このあたりの重み順位変動が、提案手法合議の意思決定に影響を及ぼしているのではないかと考えられる。

図2の対局では、複数の最善手が出るような票割れ(3:3:1など)が、全113手のうち、4回起きている。このような意見割れが起きたとき、提案手法は最も効果を発揮すると考えられる。実際、改良前よりも勝率が若干であるが上昇したので、改良がうまく働いている可能性がある。

4.3.2 局面評価値合議での票重み推移

図3に、対局中に各プレイヤーの票重みが推移する様子を示す。なお、この対局では、合議側が勝利している。

図3より、序盤での重み変動がまったく無いのがわかる。これは、序盤では評価関数を使った探索が行われず、プレイヤー票の重み付けができないためである。また、採用実績合議の表重み推移と比べると、重みの1番高いメインリーダーといえるプレイヤーが変化している様子が見てとれる。採用実績合議では、採用実績の増え方は毎回一定である。それに対して、局面評価値合議では、1手1手の局面評価値を加算していくので、合計評価値の増え方が毎回異なる。そのため、採用実績合議よりも対局中の重み順位変動がおきやすい。

票重みの加算がまったく行われていないプレイヤーがいることも見てとれる。局面評価値合議では、一回の票重み加算の処

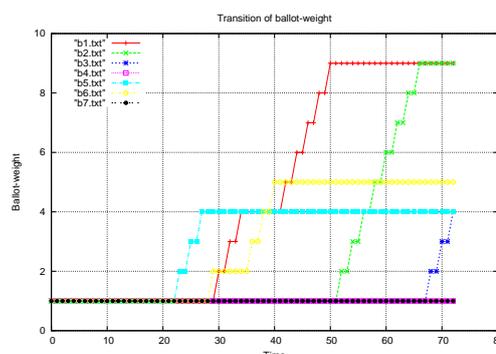


図3: 局面評価値合議での票重みの推移

理で重みが増えるプレイヤーは、合計評価値が最も高いプレイヤーのみである。重みが増えるときは、しばらく連続的に増加するが、そうでない場合は重み変動しないという傾向にある。

5. まとめと今後の課題

単純多数決による合議に、採用実績と局面評価値に基づくプレイヤー票の重み付けという改良を加えることで、採用実績合議では、改良前より高い勝率を得ることができた。しかし、現段階では、改良前対改良後の対局実験を行っていないので、改良前プログラムに対して「統計的に有意に強いのか」という検証はまだできていない。今後はこの検証も行い、改良の有効性をさらに確かめていく。

今回、プレイヤーの優劣を決定する評価方法には、採用実績と局面評価値を採用した。採用実績を使用した場合が勝率的に優れており、このとき、重み順位が2位以下のプレイヤーにのみ順位変動がみられるという特徴があることがわかった。しかし、勝率が優れていることと、このような特徴をもつということとの因果関係は明らかでない。重み順位の変動が勝率にどのような影響を及ぼすのか調査する必要がある。局面評価値合議では、重み順位が1位のメインリーダーが変動する傾向にあった。このような特徴を採用実績合議に取り入れるなどして順位変動の挙動を変化させ、勝率に与える影響を実験的に確かめていく必要がある。

参考文献

- [1] 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~", 情報処理学会研究報告, 2009-GI-22, Vol.2009, No.2, pp.1-5 (2009)
- [2] 伊藤毅志: "コンピュータ将棋における合議アルゴリズム", 人工知能学会誌, Vol.26, No.5, pp.525-530, 2011/9
- [3] "Bonanza-TheComputerShogiProgram"
http://www.geocities.jp/bonanza_shogi/
- [4] "shogi-server" <http://shogi-server.sourceforge.jp/>