

2007年度 卒業論文

3人以上で対戦するインディアンポーカーの  
AIの思考パターンを形成する手法の提案

指導教員：渡辺 大地講師

メディア学部 ゲームサイエンスプロジェクト  
学籍番号 M0104004  
赤堀 将吾

2007年度 卒業論文概要

論文題目

3人以上で対戦するインディアンポーカーの  
AIの思考パターンを形成する手法の提案

メディア学部

学籍番号：M0104004

氏名

赤堀 将吾

指導  
教員

渡辺 大地講師

キーワード

不完全情報ゲーム、AI、思考パターン、  
インディアンポーカー

不完全情報ゲームのAIは、ゲームの情報が全て見えていないために相手の行動は推測しなければならず、完全情報ゲームのAIと比べて複雑になりやすく、発展途上と言える。その不完全情報ゲームのAIの研究に、ポーカーを題材にしてハッターという戦略を利用したものがある。ハッターとは、自分が負けそうな状況でもあえて強気に勝負することで相手に自分は強い手を持っていると思わせるハッターという戦略である。この戦略を使用することでAIの勝率を上げている。しかし、ポーカーの派生ルールの1つであるインディアンポーカーを3人以上で対戦したとき、ハッターが有効的でない場合がある。インディアンポーカーは通常のポーカーとは逆に自分の手札が見えず、相手の手札が見える状態で勝負する。この独特のルールにより、ハッターを行うことが対戦相手に分かってしまうことがある。本研究では、不完全情報ゲームであるインディアンポーカーを3人以上で対戦したときに、AIがこれまでの対戦相手の行動からAI自身のカードを推測して適切な行動を取る方法を提案し、AIの行動に確実性を持たせるシステムを制作する。1回のゲームを何十回、何百回の勝負としたとき、1回1回の勝負で対戦相手の得点や賭け点、カードの強さなどを記憶していき、蓄積する。その後の勝負でAIから見た対戦相手が賭けたときや降りたときの得点や賭け点とその記憶した得点や賭け点を比較し、似通っていたときにその対戦相手の行動が記憶した得点や賭け点の時と同じである可能性があるとする。そこからAI自身が持っているカードの強さが何かを推測し、その勝負でAIが勝利できる確率を求める。これによってAIの行動に確実性を持たせる。制作したシステムを搭載したAIを人間を相手にゲームで対戦させることで検証した結果、本研究で提案した手法でAIの行動に確実さが出るようになり、有効だという結果が得られた。

# 目次

第1章	はじめに	1
1.1	研究背景	1
1.2	論文構成	3
第2章	インディアンポーカーのルール	4
2.1	インディアンポーカーの概要	4
2.2	ゲームのルール	5
2.3	本研究で定めるルール	6
第3章	AIの実装	8
3.1	行動パターンの推測	8
3.2	AIの行動の決定	10
3.3	序盤で勝負を行うときの思考	12
第4章	検証	13
4.1	検証方法	13
4.2	検証の結果	14
第5章	まとめ	17
	謝辞	18
	参考文献	19
付録A章	検証で行ったゲームの結果	22

# 第 1 章

## はじめに

### 1.1 研究背景

ゲームにはそれぞれのプレイヤーがどのような状況で行動を決定したかが全てのプレイヤーに把握できる完全情報ゲーム [1][2] と、互いに相手や自分自身の手の内が完全に、もしくは一部がわからず把握できない不完全情報ゲーム [3] がある。これらのゲームを人間を相手にして行う AI の研究はコンピュータの黎明期から行われ続けており、完全情報ゲームに関しては世界チャンピオンと互角に戦えるほどの実力を持つようになってきている。その完全情報ゲームの例として、チェスや将棋がある。これらのゲームは前述したように全ての状況が把握できるため不確実性が無く、強いアルゴリズムが開発されている。チェスでは既に AI が世界チャンピオンに勝利した実績 [4][5] が幾つも出てきており、将棋でも 10 年から 20 年後には確実にプロ棋士がコンピュータに負ける時代が来ると言われている [6]。一方、不完全情報ゲームの例としてポーカー [7] やブリッジ [8] など多数のトランプゲームや麻雀があり、それらの不完全情報ゲームでの AI の研究も行われている。エージェントモデルを用いた手法 [9]、後件部を確定していない変形ルールを用いたファジィ推論を使い、対戦相手の傾向に応じてルールを修正していく手法 [10]、完全情報ゲームにおける OM サーチを拡張した方法で、敵を惑わしパートナーを迷わせないプレイを行わせる手法 [11][12]、Actor-Critic 型強化学習と相手モデル学習を組み合わせた手法 [13][14]、AND/OR 木探索として決定論的に解く手法 [15] な

どがある。しかし、この様に不完全情報ゲームでの AI の研究は行われているが、チェスや将棋などの完全情報ゲームと比べるとプロのプレイヤーを対戦相手に想定したものではなく、まだ発展途上であると言える。不完全情報ゲームは相手の手が見えないために推測しなければならず、自分の行動の決定にはどうしても不確実性が伴ってしまうので、強いアルゴリズムがなかなか開発できていなかったが、近年になってようやく AI が人間と互角に戦えるようになってきている [16][17]。

不完全情報ゲームの 1 つであるポーカーを題材にした AI の研究で、ハッタリという戦略 [18] を用いることで AI の勝率を上げる手法 [19] がある。ハッタリとは、自分が負けそうな状況でもあえて勝負から降りず強気に出ることで相手に強いと思わせたり、逆に自分が勝ちそうな状況でも弱気に出ることで相手に弱いと思わせたりする戦略である。AI が勝利する可能性を増やすには各ゲームで行われている戦略を持たせる必要があり、この研究では AI が一定の確率で現在の自分の状況とは逆の行動、すなわちハッタリを行うようになっている。このハッタリはポーカーの場合は何人で対戦しても有効だが、3人以上で対戦するインディアンポーカーというゲームだと有効でない場合が出てくる。インディアンポーカーはポーカーの派生ルールの 1 つで、通常のポーカーとは異なり対戦相手の手札が見えていて自分の手札が見えない状態で勝負する。この独特なルールにより、インディアンポーカーでは独自の駆け引きが行われる。インディアンポーカーでも 2 人で対戦するときは、見えているカードが相手のか自分のかという違いはあるが、通常のポーカーと同じようにハッタリが有効である。しかし、インディアンポーカーは 3人以上で対戦したとき、見えているカードが増えるため通常のポーカーと状況が異なってくる。対戦相手の誰か 1 人が強いカードを持っている状況で自分が賭け点を多く出したとき、その状況を他のプレイヤーが観察できるので自分がハッタリを仕掛けたことが分かっしまい、その結果ハッタリは意味を成さなくなる。

そこで、本研究では 3人以上で対戦するインディアンポーカーで、AI の行動に確実性を持たせる方法を提案し、実装を行う。なお、本研究では 1 回のゲームが

50回の勝負で構成されているとし、最終的な得点が最も多いプレイヤーが勝者であるとする。通常、インディアンポーカーでは各プレイヤーは勝負ごとに自分から見た対戦相手の中で最も強いカードの強さを見て、その勝負に勝利できるかを考える。それに対し、AIは今までに行ってきた各プレイヤーの行動やカードの強さなどの情報を記憶して行動パターンを推測し、そこからAI自身のカードの強さを推測して勝利できる確率を求めていく。AIが勝利できる確率が高ければAIは勝負を行い、低ければ勝負から降りるといったことを行い、AIの行動に確実性を持たせる。

## 1.2 論文構成

まず、第2章でインディアンポーカーのルールを説明し、第3章で本研究で実装するAIの思考方法を示す。第4章では本研究で実装したAIを使用して実際に対戦を行ってAIの有効性を検証する。最後に、第5章で本研究のまとめを述べる。

## 第 2 章

# インディアンポーカーのルール

ここでは、インディアンポーカーの概要やルールについて説明する。2.1 節ではインディアンポーカーの概要について述べ、2.2 節にてゲームのルールを述べる。

### 2.1 インディアンポーカーの概要

インディアンポーカーとは、トランプゲームであるポーカーの派生ルールの一つ [20] である。通常のポーカーは 5 枚の手札で役を作り、その 5 枚の手札で作られた役の強弱で勝負を行うが、インディアンポーカーでは 1 枚の手札の数字で勝負を行うルールが一般的に広く認知されている。また、通常のポーカーは自分の手札のみ見えており、他の対戦相手の手札は見えないという状況で対戦するが、インディアンポーカーでは自分の手札のみ見えず、他の対戦相手の手札が見えているという、通常とは逆の状況で対戦する。これにより、インディアンポーカーは通常のポーカーとは異なった独特な駆け引きを行う必要があり、ポーカーの派生ルールの中でも変わった存在となっている。通常のポーカーではカードを 1 枚ずつ配り、その都度賭け点を出していくことになっているが、インディアンポーカーでは 1 枚しか配られないため、賭け点を出すタイミングは 1 回のみとなっている。

## 2.2 ゲームのルール

本研究ではインディアンポーカーのルールを以下のように定める。また、同時にポーカーで使用されている専門用語もここで説明する。ルールは日本ポーカープレーヤーズ協会 [20] で紹介されているポーカーのルールに準じている。

まず、それぞれのプレイヤーにカードを1枚ずつ配り、プレイヤーは自分のカードを額に掲げて開示する。それからその勝負の参加費を出し、順番に賭け点を出していく。もし勝ち目がないと思ったらその勝負から降りることも出来るが、それまでにその勝負で出した賭け点は戻ってこない。この勝負から降りることを「フォールド」と言う。勝負するときには賭け点を出して参加する意思を見せるが、このときプレイヤーは前のプレイヤーが賭けた金額と同額を出すか、それ以上の金額を出すかを選択できる。前のプレイヤーと同じ額を出すことを「コール」、前のプレイヤーよりも高い額を出すことを「レイズ」と言う。また、まだ誰も賭け点を出していないときに限り、賭け点を出さず次の人に回すという行動を取ることが出来る。この次の人に順番を回すことを「パス」と言う。ただし、誰か一人でもかけていた場合はパスを行うことが出来ない。そして、カードを提示して勝負する時には勝負を行うプレイヤー全員の賭け点が同じになっていなければならない。なので、一巡して参加している全てのプレイヤーが賭け点を出したとしても、全員の賭け点が同じになるまで賭け続けていかなければならない。全員が同じ額を提示したら、額に当てていたカードを自分にも見えるように提示し、結果を見る。提示されたカードの中で最も強いカードを持っていた者がその勝負での勝者となり、その勝負での賭け点を全て得ることが出来る。これを一定回数行い、最終的に最も得点が多いプレイヤーがそのゲームの勝者となる。図2.1は、1回の勝負の流れをチャートとして表したものである。



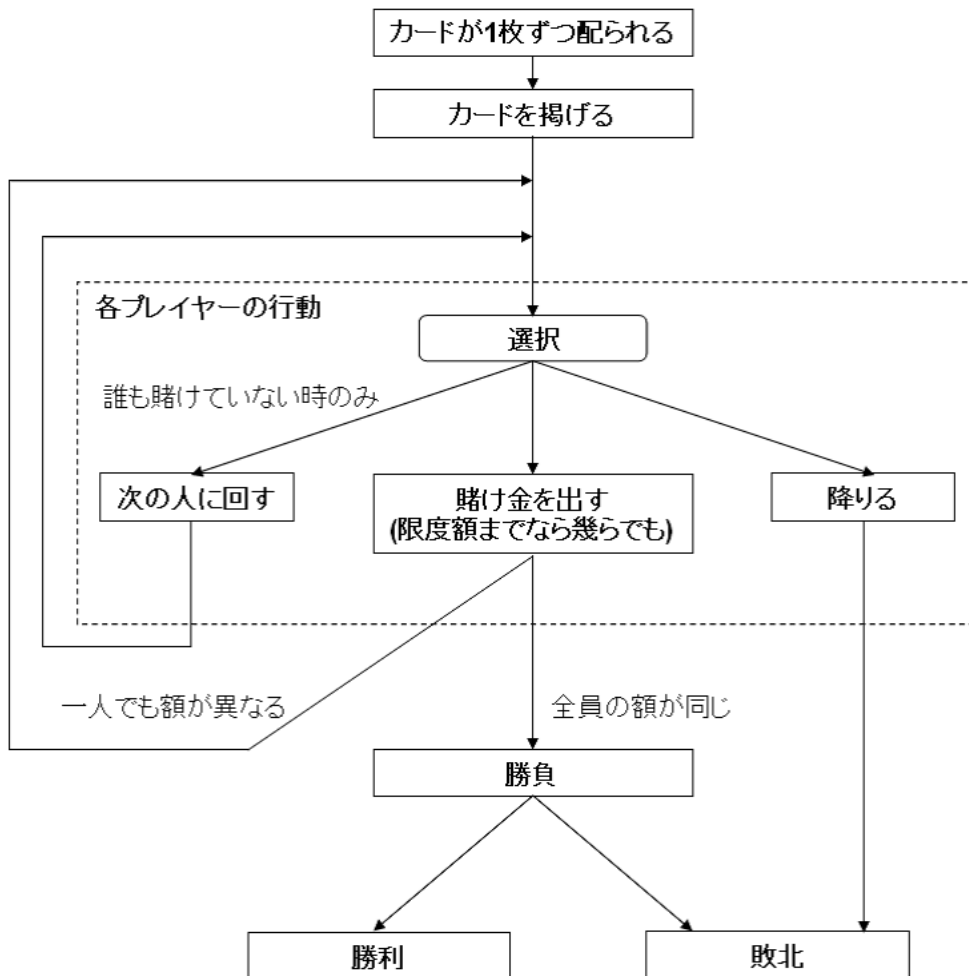


図 2.1: 1 回の勝負の流れ

## 2.3 本研究で定めるルール

ここでは本研究で AI を実装するために独自に定めたルールについて説明する。ゲームは 50 回の勝負を 1 ゲームとし、プレイする人数は 3 人とする。また、ゲーム開始時の得点は 10000 で、1 回の勝負で賭けられる賭け点の上限は現在の得点までとする。もしゲームの途中で得点を全て失ったら、借金のように資金 10000 を得て続行し、ゲーム終了時に最終的な得点から借金した分を差し引く。また、他のプレイヤーが自分の得点以上を出してきたら賭けることが出来ないのも、その場合のみ提示された額で勝負することが出来る。ただし、その勝負で負けたら同

じように 10000 を得て足りない差額を払うことになる。勝負の開始時には参加費として 100 支払う。勝敗を決めるためのカードの強弱は、わかりやすくするために最も弱いのが 1 で最も強いのが 13 と数字の大きさをそのまま強弱とする。カードは 1 から 13 まで各 1 枚ずつ計 13 枚のカードの組を使用する。これにより、引き分けは存在せず必ず勝敗が出るようにしている。そして、1 回の勝負が終了するたびに使用したカードを回収してシャッフルしたものを再び使用する。ゲーム終了時に最も得点が多かったプレイヤーがそのゲームでの勝者となる。

# 第 3 章

## AIの実装

ここでは本研究で制作する AI の実装方法について説明する。まず 3.1 節で情報の蓄積と AI から見た対戦相手の行動パターンを推測する方法、3.2 節でゲームの中盤以降で AI が推測した対戦相手の行動パターンから AI 自身のカードを推測する方法を説明する。また、3.3 節ではゲームの序盤で AI が行動を決定する方法を説明する。なお、本研究では 1 回のゲームを 50 回の勝負としているので、最初の 15 回の勝負を序盤、残りの 35 回を中盤以降とする。

### 3.1 行動パターンの推測

ゲームの序盤は、AI から見た対戦相手の状態や行動といった情報の収集を毎回の勝負が終了した後に行う。収集する情報は以下の通りである。

- カードの強さ
- 勝負終了時点での得点
- 調べている対象のプレイヤーが降りていた場合はそのプレイヤーが降りていたときに場に提示されていた賭け点、プレイヤーが賭けていた場合は勝負が終了したときに場に提示されていた賭け点
- 降りたか賭けたか

これらの収集した情報をもとに、AI から見た対戦相手の行動を推測する。通常、インディアンポーカーでプレイヤーが行動を決定するのは、第一に対戦相手のカードの強さ、その次に自分の得点や現在場に提示されている賭け点である。例えば、あるプレイヤーから見た対戦相手が持っているカードのうち最も強いカードの強さが12であったとすると、通常ならば勝つ可能性が低いので降りることを選択するが、ゲームの終盤で相手と自分の得点に大差がつけられて負けている状況であれば、一発逆転を狙って勝負に出ることを選択する場合もある。このように各プレイヤーはカードの強さ、得点、賭け点で賭けるか降りるかを決定する。つまり、対戦相手の行動を記憶し、それから後の勝負で同じような行動を取っていたら自分のカードが何かを推測することが出来る。そこで、AI から見た対戦相手のうち任意の相手プレイヤーをプレイヤー X とし、AI はプレイヤー X に対してプレイヤー X が勝負を降りたか賭けたかという行動と、プレイヤー X から見た対戦相手の持っているカードのうち最も強いカードの強さ、そして以下に挙げる値を関連する情報として記憶する。なお、プレイヤー X から見た対戦相手とは AI とプレイヤー X でないもう1人のプレイヤーである。

1. プレイヤーの現在の得点と AI を含む他のプレイヤーの得点のうち最も多い得点の差
2. 賭けていたときは勝負が終了したときに場に提示されていた賭け点
3. 降りていたときは降りたときに場に提示されていた賭け点
4. 賭け点をプレイヤーが持っている得点で割った値

1 番目はプレイヤー X の現在の得点を  $c$ 、プレイヤー X 以外で最も得点が多いプレイヤーの現在の得点を  $C$  として (3.1) で求めた値  $y$  である。また、4 番目はプレイヤー X が降りていた場合はプレイヤー X が降りたときに場に提示されていた賭け点、プレイヤー X が賭けていた場合は勝負が終了したときに場に提示されていた賭け点を  $b$  とし、プレイヤー X の得点を  $c$  として式 (3.2) で求めた値  $w$  であ

る。これらを AI から見た対戦相手 2 人行って記憶し、蓄積していく。

$$y = C - c \quad (3.1)$$

$$w = \frac{b}{c} \quad (3.2)$$

## 3.2 AIの行動の決定

ゲームの中盤から、AIは序盤にて推測した各プレイヤーの行動パターンから AI 自身のカードの強さを推測して行動を決定する。また、中盤以降もゲーム内の情報を記憶して情報量を増やしていく。

はじめに、AI から見た対戦相手やゲーム中に得られる情報をここで定義する。AI から見た 2 人の対戦相手のうち、最も強いカードを持っているプレイヤーをプレイヤー A、残るもう 1 人のプレイヤーをプレイヤー B とし、プレイヤー A が持つカードの強さを  $N_A$ 、プレイヤー B が持つカードの強さを  $N_B$  とする。AI に順番が回ってきたら、まずプレイヤー A が行動していたか、行動していたら賭けていたか降りていたかを調べる。次にプレイヤー A が降りていた場合はプレイヤー A が降りていたときに場に提示されていた賭け点を、プレイヤー A が賭けていた場合は勝負が終了したときに場に提示されていた賭け点を  $b$  とし、プレイヤー A の得点を  $c$  として式 (3.2) で  $w$  を求める。そして、プレイヤー B と AI の 2 人のうち最も得点が多いプレイヤーの得点を  $C$  として式 (3.1) で  $y$  を求める。プレイヤー A が降りたか賭けたかという行動と求めた  $y$  と  $w$ 、得点の値を記憶した値と比べ、それぞれの値のいずれかに近い値と関連づけられたカードの強さを全て記憶する。近い値というのは、 $y$  や賭け点なら求めた値からプラスマイナス 500 の範囲、 $w$  ならプラスマイナス 0.05 の範囲である。値が完全に一致するということは可能性が非常に低いので、このように範囲を取ることでそれまでにあった状況に近いと判断し選択するのである。最後に記憶したカードの総数を  $m$  とし、プレイヤー A が

降りていた場合は記憶したカードの強さのうち  $N_B$  を上回る強さを持つカードの総数を  $n$ 、プレイヤー A が賭けていたときは  $N_A$  を上回る強さを持つカードの総数を  $n$  として式 (3.3) で勝利できる確率  $P$  を求める。ここで求めた確率  $P$  は  $P_A$  として記憶する。プレイヤー A が行動していなかった場合はこれらを行わず、プレイヤー B の行動からのみ判断する。

$$P = \frac{n}{m} \quad (3.3)$$

プレイヤー A の行動から確率  $P_A$  を求めたら、プレイヤー B でも同じように AI が勝利できる確率  $P_B$  を求める。ただし、 $n$  はプレイヤー A の時と同じようにプレイヤー A が降りていた場合は記憶したカードの強さのうち  $N_B$  を上回る強さを持つカードの総数を、プレイヤー A が賭けていたときは  $N_A$  を上回る強さを持つカードの総数を  $n$  とする。そして、求めた  $P_A$  と  $P_B$  の平均値を最終的な確率  $P_f$  とし、 $P_f$  が 0.5 以上であれば勝負を行い、0.5 未満であれば降りることを選択する。これは序盤での行動の決定とは異なり、得点を確実に増やすために堅実に勝負を行うためである。勝負を行うときに AI が提示する賭け点はその提示する賭け点を  $B$ 、AI から見た対戦相手のうち最も得点が多いプレイヤーの得点を  $C$ 、AI 自身の得点を  $c$  として式 (3.4) で求める。ただし、 $B$  の値が 0 を下回っていた、つまり AI が最も得点を持っている場合は賭け点を上げずに勝負する。

$$B = \frac{C - c}{2} \quad (3.4)$$

$P_A$  か  $P_B$  のいずれかが求められていない場合は求められている方の確率のみで判断を行う。プレイヤー A とプレイヤー B が共に行動していなかった場合は、パスが可能であれば AI はパスを選択して次のプレイヤーに順番を回す。パスが不可能であった場合は AI は自身のカードの強さを推測できないので、3.3 節で示した方法を使用して行動を決定する。また、プレイヤー A とプレイヤー B が共に降りていた場合は自動的に AI の勝利となるので推測は行わない。

### 3.3 序盤で勝負を行うときの思考

ゲームの序盤ではまだ記憶した情報をもとに AI から見た対戦相手の行動パターンを分類できていないため、対戦相手のカードの強さから行動を決定する。

AI から見た対戦相手のうち、最も強いカードを持っているプレイヤーのカードの強さを  $n$  とする。ただし、そのプレイヤーが降りていた場合はもう 1 人のプレイヤーが持つカードの強さを  $n$  とする。そして式 (3.5) を用いて AI がその勝負で勝利できる確率  $p$  を求める。

$$p = \frac{13 - n}{11} \quad (3.5)$$

$p$  が 0.25 以上であった場合、つまり勝利できる確率が 25 パーセントを超えていた場合は必ず降りずに勝負を行う。式 (3.5) で  $p$  が 0.25 以上となるのは  $n$  が 10 以下のときなので、最も強いカードが 10 以下であれば必ず勝負を行うということになる。通常は 8 以上になると勝利できる確率が 5 割を切り、負ける確率の方が高くなる。しかし、実際にゲームを行ったとき、5 割を切った時点で降りる可能性を出すことで降りることによる損失が多くなってしまいうので必ず勝負を行う状況を多くして強気な行動をさせて降りることによる損失を減らしている。 $p$  が 0.25 未満であった場合、 $p$  をそのまま勝負を行う確率にし、0 から 1 までの乱数で出した値が  $p$  未満であれば勝負を行い、 $p$  以上であれば降りることを選択する。なお、勝負を行うと決定したときの賭け点はカードの強さにかかわらず賭け点を上げないで勝負を行い、常に賭け点を AI 自身が決めないようにする。これは AI の行動から対戦相手が対戦相手自身のカードを推測することを防ぐためである。

# 第 4 章

## 検証

### 4.1 検証方法

検証の目的は、本研究で提案した手法が実際に人間と対戦し、勝利することで本研究の手法の有効性を証明するためである。検証では実際に人間を相手にゲームを行う。AIを含めて3人で対戦し、50回の勝負を1ゲームとする。図4.1は本研究で実装したAIがゲームの状況を知るためのアプリケーションである。

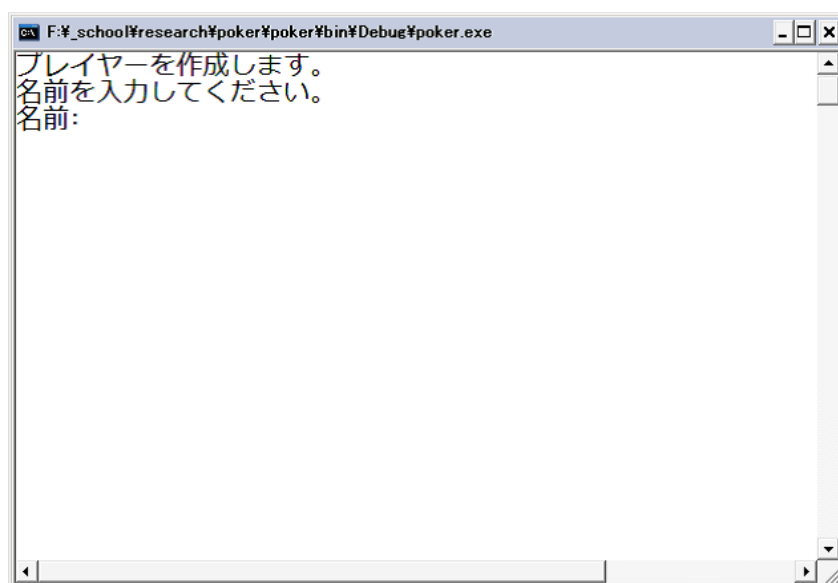


図 4.1: AI が情報を知るためのアプリケーション



AIは思考のみを行い、実際のプレイは代理の人間のプレイヤーが行うようになっている。代理のプレイヤーがアプリケーションを使用して行う手順は以下の通りである。

1. 対戦相手の名前を入力する
2. 全プレイヤーの行動する順番を入力する
3. 配られたカードの強さを AI 自身のものを除いて入力する
4. 対戦相手の行動を入力する
5. AI の順番になったら AI が自身の行動を示すので、代理プレイヤーはそれに従い行動する
6. 全員の賭け点と同じになるまで 4 から繰り返す
7. 賭け点が同じになったら AI 自身のカードの強さを入力する
8. 3 に戻って繰り返す

代理のプレイヤーはこの手順通りに情報を入力していき、AIの行動を代理で行う。なお、手順4では対戦相手が降りたか賭けたか、賭けたのならば賭け点を幾ら出してきたのかを入力する。また、手順7ではAI自身のカードの強さを入力した後に誰がその勝負で勝利したのかを自動で判別し、AIも含む全てのプレイヤーの得点の情報が賭け点に応じて自動的に増減するようにしている。

今回の検証では、10回のゲームを合計20人に対して行った。各プレイヤーの実力はルールが分かる程度である。付録Aに全ゲームの得点推移を示す。

## 4.2 検証の結果

検証の結果、本研究の提案した方法でAIは10回のゲームで5勝5敗という成績を収めた。表4.1は各ゲーム終了時のAIの得点である。得点がマイナスになっ

ているところもあるが、これは第2章で示したように得点がマイナスになってもプレイが進行できるルールを定めたからである。

表 4.1: ゲームの結果

ゲームの回数	AIの得点
1	49200
2	-2400
3	-7000
4	37600
5	-12000
6	17200
7	35200
8	-20400
9	28500
10	-700

本研究で提案した手法によって損失を少なくすることが出来た場面があった。表 4.2 は 4 ゲーム目での 41 回目の勝負の結果である。表 4.2 で示している開始プレイヤーの項目は、その勝負を始めたときに最初に賭けるか降りるかを決めたプレイヤーである。プレイヤーの項目は各プレイヤーの名前、所持カードの項目は各プレイヤーがその勝負で所持していたカードの強さ、得点の項目はその勝負が終了して得点を受け渡しした後の各プレイヤーの得点、行動の項目は各プレイヤーがその勝負で賭けたか降りたかを示したものである。賭け点の項目は、プレイヤーが降りたときはそのときにプレイヤーが支払った点数、賭けたときはカードを提示するとき場に提示されていた賭け点である。この勝負は序盤での行動を決定する方法ならば勝負を行っているところだが、AI は降りることで得点の損失を最小に減らしている。これはそれまでの 40 回の勝負で人間 2 が賭け点を 5000 にしたとき、人間 2 から見た AI も含む対戦相手が持つカードの中で最も強いカードの強さのほとんどが、今回の勝負で人間 2 が持っているカードの強さである 7 を下回っており、AI は今回の勝負でも負けると推測したから行った行動である。実際、AI が持っていたカードの強さは 1 であり、そのまま勝負していれば AI は負けていた。

表 4.2: 4 ゲーム目での 41 回目の勝負の結果

勝負	開始プレイヤー	プレイヤー	所持カード	得点	賭け点	行動
41	人間 2	AI	1	20200	100	降りた
		人間 1	7	7100	5000	賭けた
		人間 2	4	2700	5000	賭けた

しかし、この行動決定によって逆に勝機を逃してしまった場面もあった。表 4.3 は 1 ゲーム目での 31 回目の勝負の結果である。表 4.3 で示している各項目は表 4.2 と同じである。31 回目の勝負では実際には AI が勝利出来る状況であったが、AI は降りてしまっている。これはそれまでの 30 回の勝負で人間 1 や人間 2 が行ってきた行動と異なっており、AI が正確に推測できなかったためである。対戦した人間のプレイヤーが出す賭け点が特定の値に偏り、かつその特定の値それぞれの差が大きく開いていたため AI がうまく推測できなかった。

表 4.3: 1 ゲーム目での 31 回目の勝負の結果

勝負	開始プレイヤー	プレイヤー	所持カード	得点	賭け点	行動
31	人間 1	AI	12	27200	100	降りた
		人間 1	1	-200	4000	賭けた
		人間 2	11	3000	4000	賭けた

## 第 5 章

### まとめ

本研究では、ゲーム内で得られる情報からプレイヤーがどのような行動パターンを持っているのかを推測し、推測した結果から AI 自身のカードの強さを推測する手法を提案した。この手法は、人間のプレイヤーが行った行動により得られたそれぞれの値がカードの強さ毎にある程度決まった値になっている、つまり人間のプレイヤーが AI 自身のカードの強さを推測できるような行動を行ったときに有効であることが検証の結果からわかった。

今後はどのような場合でも同じように賭け点を出すといった、行動がワンパターンである人間のプレイヤーに対しては行動パターンからの推測は行わずに、本研究でゲームの序盤に AI が行ったような勝利できる確率から行動を決定する方法を使うなど、プレイヤー毎に戦略を変えていく手法を提案していきたい。

# 謝辞

本研究を進めるにあたり、多くのご指導とご教授をいただきました渡辺大地講師、および三上浩司氏、中村太戯留氏、小澤賢侍氏に感謝の意を表します。また、共に学び、助力しあい、励ましあってきた友人達にも深く感謝いたします。

最後に、温かく見守ってくれた家族に感謝いたします。

## 参考文献

- [1] 鈴木光男, ”ゲーム理論入門”, 共立出版株式会社, 2003.
- [2] John von Neumann and Oskar Morgenstem, ”Theory of Games and Economic Behavior”, John Wiley & Sons, Inc., New York, 1944.
- [3] 作田誠, ”不完全情報ゲームの研究”, オペレーションズ・リサーチ:経営の科学 Vol.52, No.1, pp.27-34, 2007.
- [4] ChessBase.com - Chess News - Kramnik vs Deep Fritz: Computer wins match by 4:2, <http://www.chessbase.com/newsdetail.asp?newsid=3524>.
- [5] ブルース・パンドルフィーニ, 鈴木知道, ”ディープブルー vs. カスパロフ”, 河出書房新社, 1998.
- [6] 松原仁, ”コンピュータ将棋はどのようにしてアマ5段まで強くなったか”, 情報処理 Vol46, No.7, pp.814-816, 2005.
- [7] 日本ポーカープレイヤーズ協会 - ポーカーについて, <http://www.japan-poker.net/modules/xfsection/>.
- [8] 上原貴夫, ”コンピュータブリッジ”, 人工知能学会誌 Vol.16, No.3, pp.385-392, 2001.

- [9] D.Billings, D.Papp, J.Schaeffer, and D.Szafron, "Opponent Modeling in Poker", Proceedings of AAAI-98/IAAAI-98, 1998, pp.493-499.
- [10] 鬼沢武久, 風見覚, 高橋千晴, "不完全情報ゲームプレイングシステムの構築 : スタッドポーカーを例にして", 日本知能情報ファジィ学会誌 Vol.15, No.1, pp.127-141, 2003.
- [11] 小和田友仁, 上原貴夫, "コンピュータブリッジにおける他者のモデルを考慮したゲーム木探索の提案", 情報処理学会論文誌 Vol.47, No.11, pp.3005-3016, 2006.
- [12] 小和田友仁, 埜敏博, 上原貴夫, "コンピュータブリッジのプレイヤーモデルに基づく並列ゲーム木探索", 情報処理学会研究報告 Vol.2006, No.70, pp.75-82, 2006.
- [13] 松野陽一郎, 山崎達也, 松田潤, 石井信, "相手学習に基づくマルチエージェントゲームの強化学習", 電子情報通信学会技術研究報告, Vol.100, No.688 pp.91-98, 2001.
- [14] 藤田肇, 松野陽一郎, 石井信, "マルチエージェントカードゲームのための強化学習法の改良", 電子情報通信学会技術研究報告, Vol.102, No.731 pp.167-172, 2003.
- [15] 作田誠, 飯田弘之, "不確定性を持つ問題を解くための AND/OR 木探索: 衝立詰将棋を題材として", 情報処理学会論文誌 Vol.43, No.1, pp.1-10, 2002.
- [16] 人間対コンピュータのポーカー対決、勝者は人間 - CNET Japan, <http://japan.cnet.com/news/biz/story/0,2000056020,20353447,00.htm>.
- [17] Man vs. Machine Poker Challenge? In conjunction with AAAI-07 <http://www.aaai.org/Pressroom/Releases/release-07-0612.php>.

- [18] PokerTips.org - ポーカー攻略法：心理作戦, <http://www.pokertips.jp/strategy/deception.php>.
- [19] 高橋千晴, 鬼沢武久, ”不完全情報ゲームにおける意志決定プロセス”, 日本機械学会第12回インテリジェント・システム・シンポジウム講演論文集, No.02-10, pp.297-302, 2002.
- [20] 日本ポーカープレイヤーズ協会 - ポーカーについて-ポーカーの種類, <http://www.japan-poker.net/modules/xfsection/article.php?articleid=3>.



# 付録 A

## 検証で行ったゲームの結果

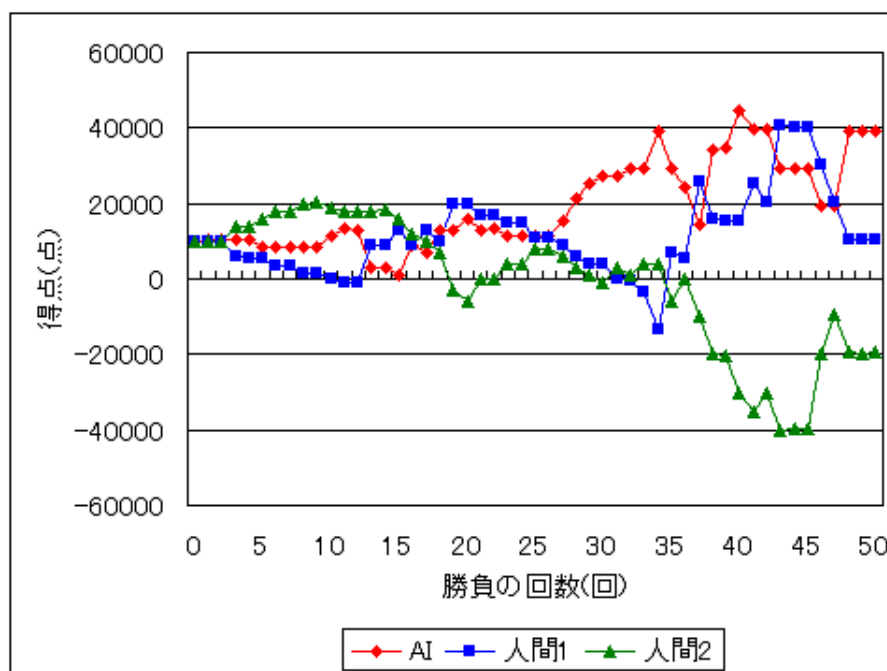


図 A.1: 1 回目のゲーム

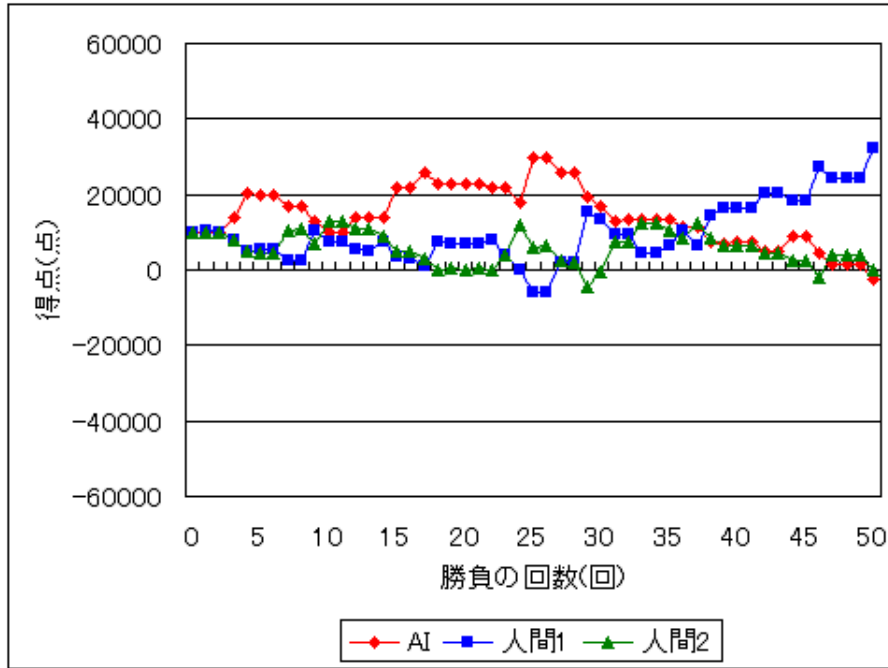


図 A.2: 2 回目のゲーム

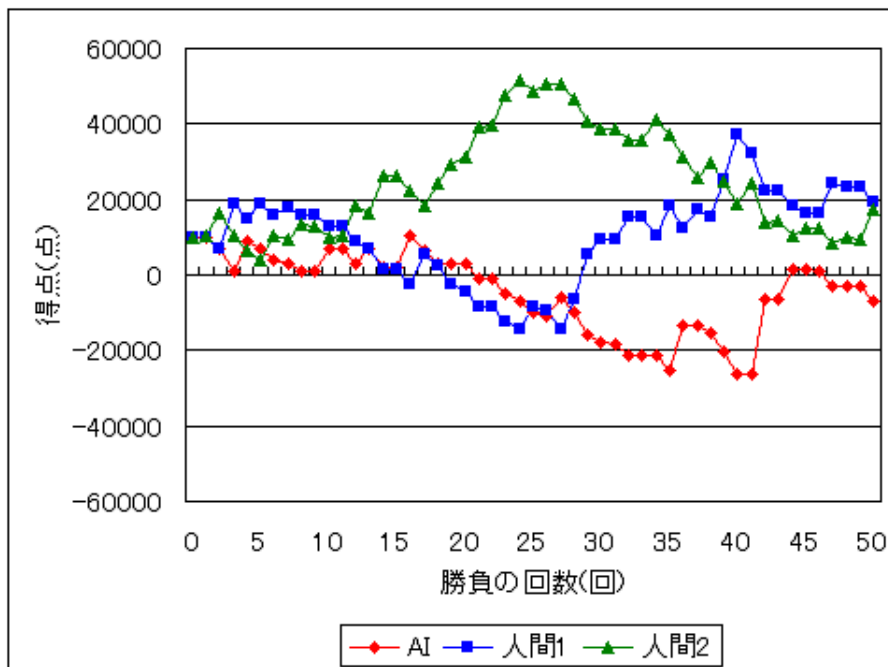


図 A.3: 3 回目のゲーム

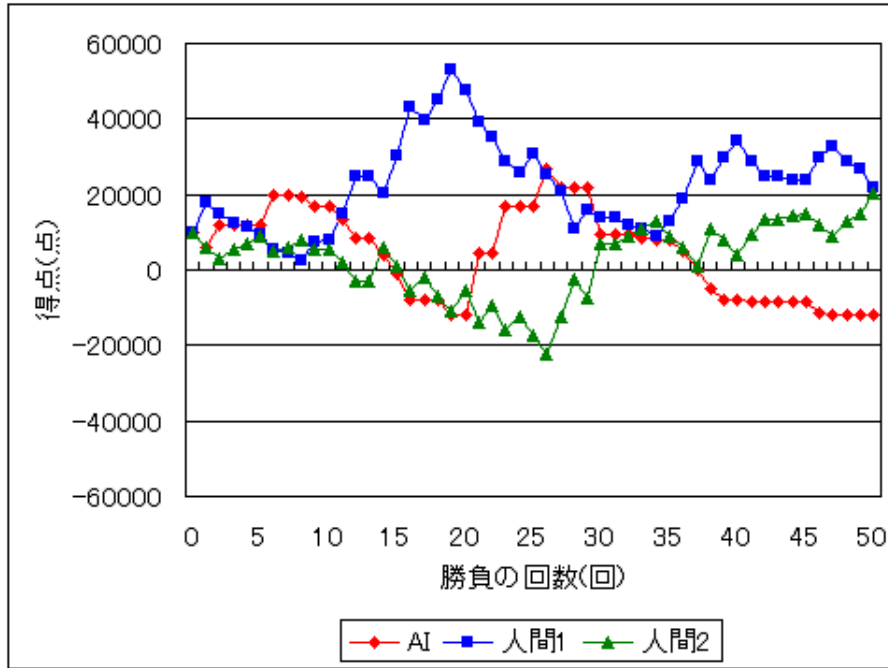


図 A.4: 4 回目のゲーム

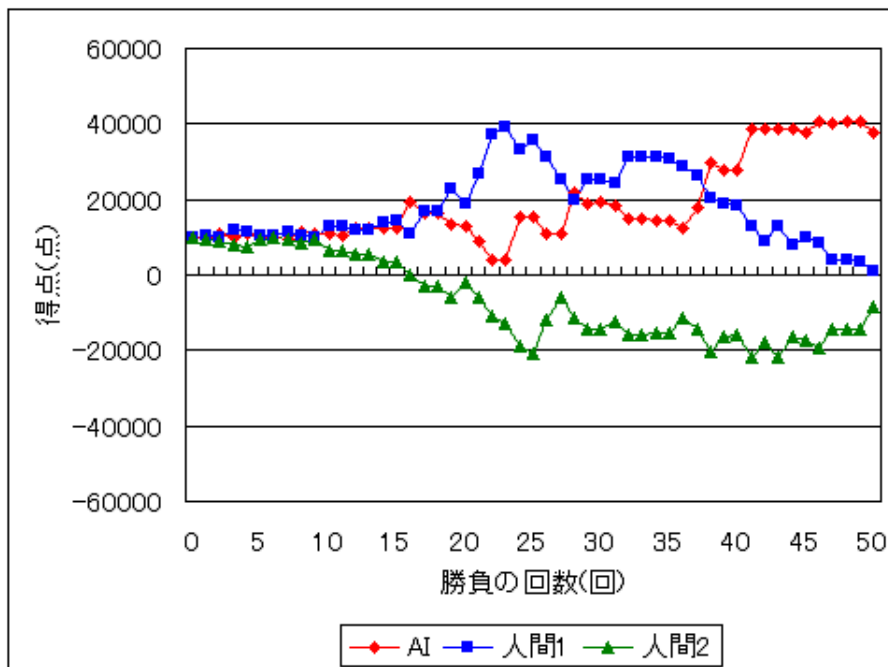


図 A.5: 5 回目のゲーム

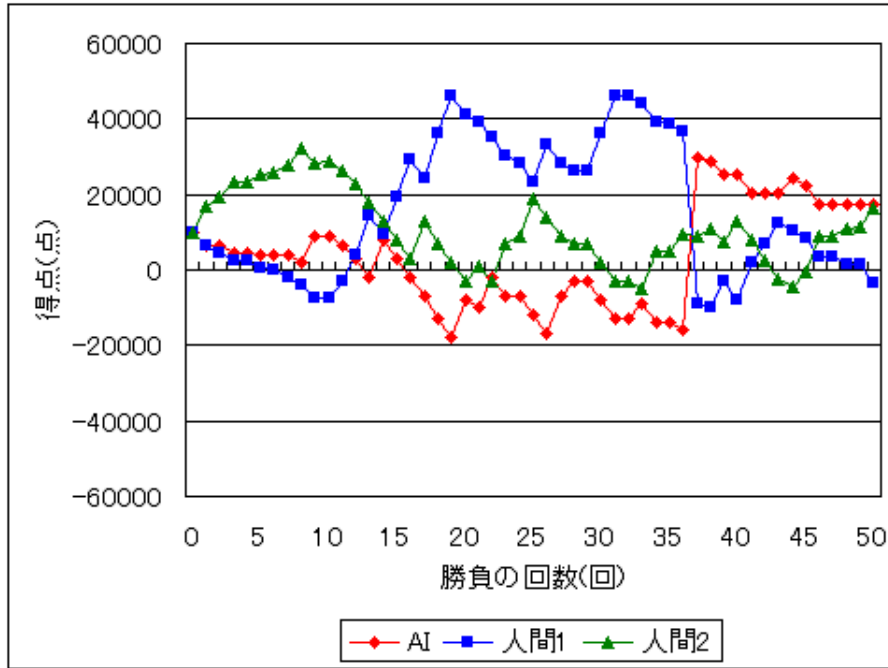


図 A.6: 6 回目のゲーム

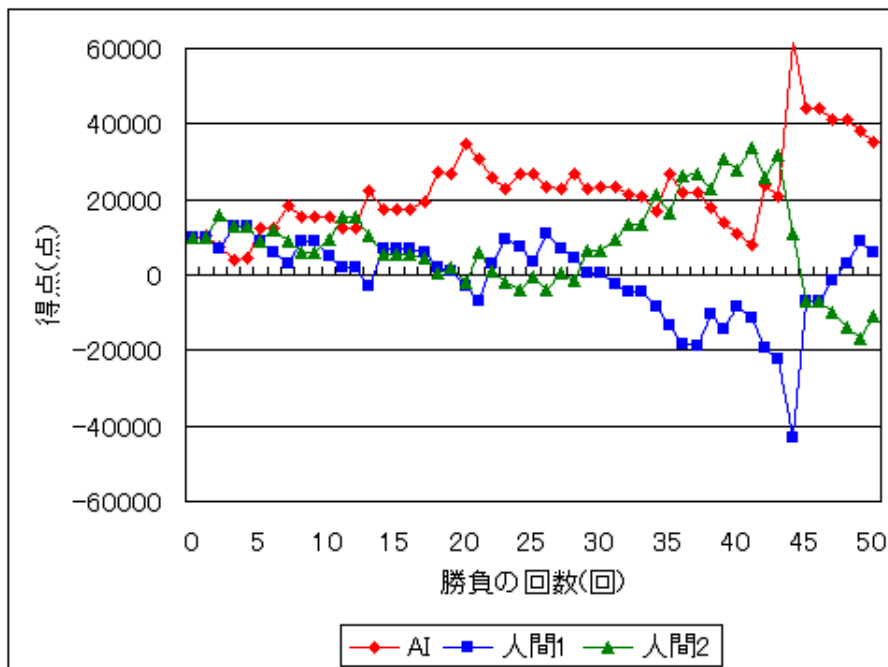


図 A.7: 7 回目のゲーム

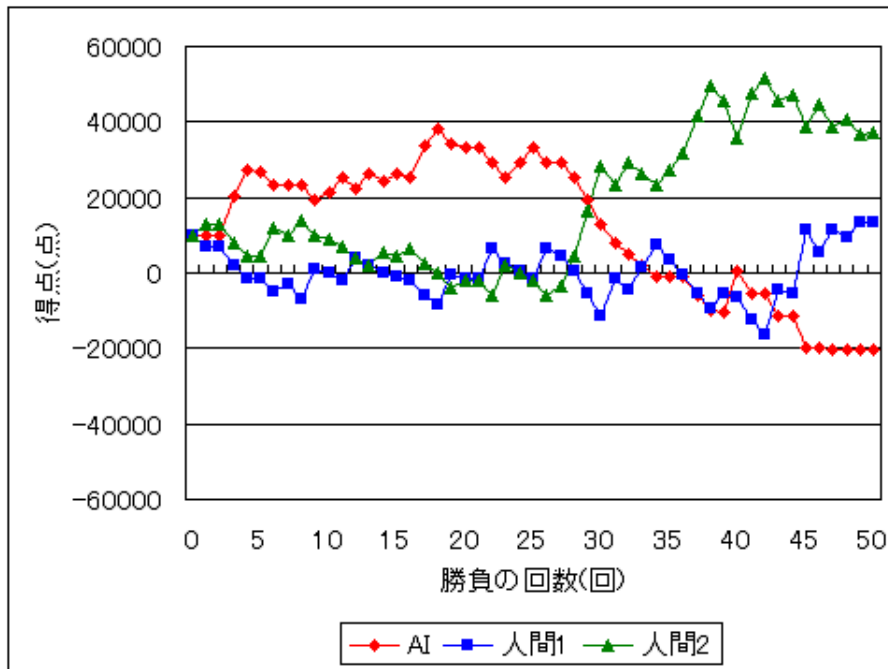


図 A.8: 8 回目のゲーム

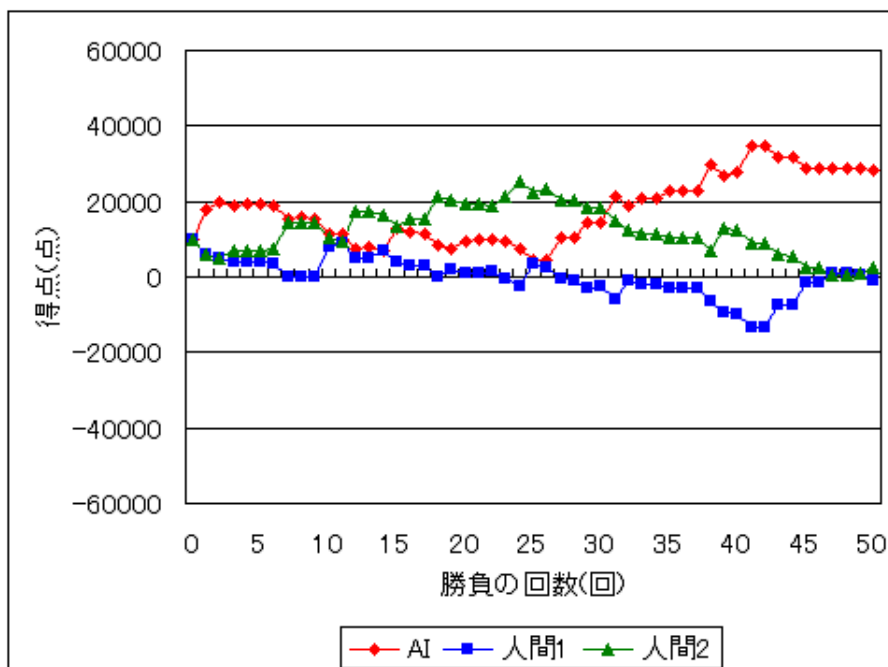


図 A.9: 9 回目のゲーム

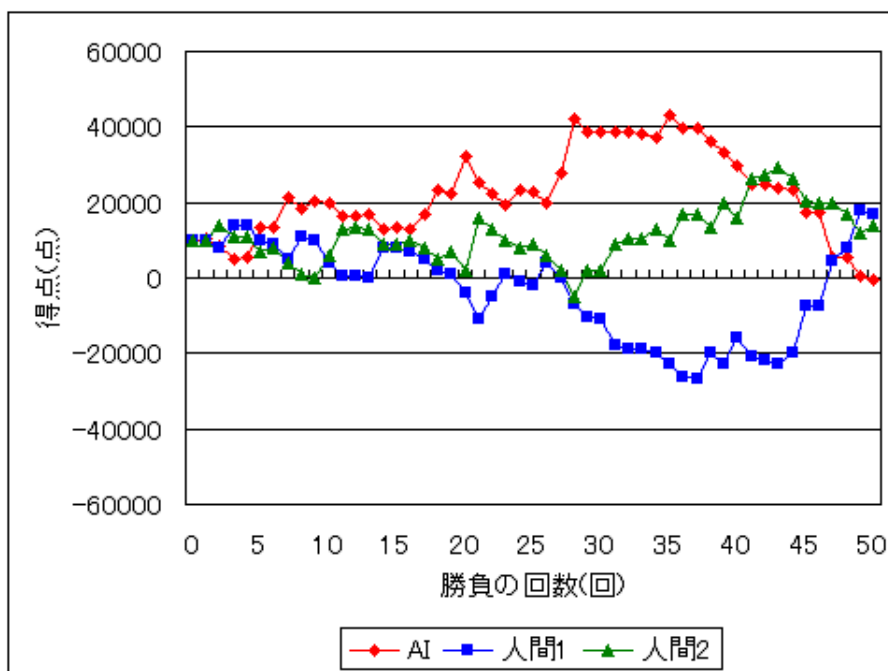


図 A.10: 10 回目のゲーム