

令和 6 年度
修士学位論文

相手手札推定を動的に切り替える 大貧民プレイヤーの研究

**A Study on DAIHINMIN Players That Dynamically Switch
Opponent-Hand-Estimation Methods**

1275108 中山 友里歌

指導教員 松崎 公紀

高知工科大学大学院 工学研究科 基盤工学専攻
情報学コース

要 旨

相手手札推定を動的に切り替える大貧民プレイヤーの研究

中山 友里歌

不完全情報ゲームにおいて、見えない情報の推測は重要であると考えられている。不完全情報ゲームのひとつである大貧民では、相手手札推定の精度がプレイヤーの強さに正の影響があることが示されている。これまでに推定精度の高い相手手札推定手法が複数提案されているが、優れた推定手法は一般に計算コストが増大することから最も優れた推定手法を一概に決定することは難しい。本研究では、相手手札推定をより効果的に行うためのアプローチとして、事前に用意した複数の相手手札推定手法を対戦中に切り替えることを提案する。相手手札を完全に知るチートを含めた5つの相手手札推定手法を切り替えるプレイヤーを作成し、2種類のプレイヤーを用いて評価実験を行った。その結果、用意した相手手札推定手法のうち最も適切なものに収束するという結果が得られた。

キーワード 大貧民, 大富豪, 相手手札推定, バンディットアルゴリズム

Abstract

A Study on DAIHINMIN Players That Dynamically Switch Opponent-Hand-Estimation Methods

NAKAYAMA, Yurika

In incomplete information games, the estimation of hidden information is considered important. In DAIHINMIN, an incomplete information game, it has been shown that the accuracy of the opponents' hand estimation positively affects a player's strength. Several high-accuracy estimation methods have been proposed; however, more accurate methods generally require higher computational cost, making it difficult to universally determine the best method. In this study, we propose an approach to improve opponents' hand estimation by dynamically switching between several pre-prepared estimation methods during the game. We implemented a player that switches between five estimation methods, including a cheating method that fully knows the opponents' hands, and conducted evaluation experiments with two types of players. As a result, the player converged to the most appropriate estimation method among those available.

key words DAIHINMIN, DAIHUGO, Opponent-hand Estimation, Bandit Algorithms

目次

第 1 章	はじめに	1
第 2 章	コンピュータ大貧民	3
2.1	コンピュータ大貧民のルール	3
2.2	コンピュータ大貧民の既存クライアント	4
2.2.1	Blauweregen	5
2.2.2	Ninja	5
2.2.3	switch2	5
2.2.4	PoliXLite	5
2.3	相手手札推定手法	6
第 3 章	提案手法	9
第 4 章	実験	11
4.1	評価対象プレイヤーが 1 つの場合の評価	11
4.2	試合開始時の階級を考慮した評価値を用いる効果の評価	17
4.3	対戦相手によって推定精度が大きく異なる場合の評価	20
第 5 章	まとめ	25
	謝辞	26
	参考文献	27

目次

3.1	相手手札推定アルゴリズムの切り替え方法	9
4.1	実験 1-S-C (switch2, CHEAT あり) における推定方法累積選択割合の推移	14
4.2	実験 1-S-K (switch2, CHEAT なし) における推定方法累積選択割合の推移	14
4.3	実験 1-P-C (PoliX Lite, CHEAT あり) における推定方法累積選択割合の推移	15
4.4	実験 1-P-K (PoliX Lite, CHEAT なし) における推定方法累積選択割合の推移	15
4.5	実験 1-M-C (2 種類のプレイヤー, CHEAT あり) における推定方法 累積選択割合の推移	16
4.6	実験 1-M-K (2 種類のプレイヤー, CHEAT なし) における推定方法 累積選択割合の推移	16
4.7	実験 2-PT-K4 (得点のみ, Kunoichi×4) における推定方法 累積選択割合の推移	18
4.8	実験 2-WR-K4 (階級考慮, Kunoichi×4) における推定方法 累積選択割合の推移	18
4.9	実験 2-PT-K3 (得点のみ, Kunoichi×3) における推定方法 累積選択割合の推移	19
4.10	実験 2-WR-K3 (階級考慮, Kunoichi×3) における推定方法 累積選択割合の推移	19
4.11	$n = 50$ における各推定方法の累積選択割合の推移	22
4.12	$n = 60$ における各推定方法の累積選択割合の推移	22
4.13	$n = 70$ における各推定方法の累積選択割合の推移	23
4.14	$n = 80$ における各推定方法の累積選択割合の推移	23
4.15	$n = 90$ における各推定方法の累積選択割合の推移	24
4.16	$n = 100$ における各推定方法の累積選択割合の推移	24

表目次

2.1	各推定手法を用いた改造 Blauweregen の獲得得点	8
2.2	各推定手法の推定精度	8
4.1	実験 1 における評価プレイヤーと Blauweregen の獲得得点平均	13
4.2	実験 1-S-C, 実験 1-P-C, 実験 1-M-C (CHEAT あり) において, 各相手手 札推定手法が選択された回数と, その試合での獲得得点の平均 (括弧内) . .	13
4.3	実験 1-S-K, 実験 1-P-K, 実験 1-M-K (CHEAT なし) において, 各相手手 札推定手法が選択された回数と, その試合での獲得得点の平均 (括弧内) . .	13
4.4	実験 2 における評価プレイヤーの獲得得点合計	20
4.5	実験 3 における評価プレイヤーの獲得得点平均と CHEAT の推定精度	21

第 1 章

はじめに

ゲーム情報学では、囲碁や将棋，チェスなどの完全情報ゲームにおいて優れたコンピュータプレイヤーが実現されており [1, 2]，近年は不完全情報ゲームに関しても研究が進んでいる．近年では，ポーカーや麻雀，大貧民といった不完全情報ゲームに対する関心も高まり，多数のプレイヤーが開発されている．本研究の対象とする「大貧民」は多人数不完全情報ゲームの一種であり，UEC コンピュータ大貧民大会（UECda） [3] を通して多数のコンピュータプレイヤーが作成されてきた．

UECda における上位クライアントの多くはモンテカルロ法で実装されている．モンテカルロ法を行うにあたっては，各プレイヤーから見えない相手手札を何らかの方法で補完し，その情報を基にしてシミュレーションを行うのが一般的である．そこで補完する相手手札情報をより正確に推定することはコンピュータ大貧民プレイヤーの強さに正の影響があると考えられている．実際著者の先行研究 [9] において，相手手札情報を正しく推定する割合（推定精度）がプレイヤーの強さに正の影響があることだけでなく，弱い手札を推定できることがより大きな効果があることを確認した．

大貧民における相手手札の推定手法については，機械学習や深層学習を利用する方法を中心に，複数の手法が提案されている [11, 5, 6, 10, 12]．これらの手法は一般に，推定精度を高くするためにより多くの計算量を必要とする．特にモンテカルロ法のようにシミュレーション回数も重要なアルゴリズムにおいては，推定精度と計算コストはトレードオフの関係がある．したがって，著者らが知る限りでは，モンテカルロ法と組み合わせた際に最適と言いつけることのできる相手手札推定手法は現段階では確立されていない．

本研究では，優れた相手手札推定手法を開発するのではなく，既存の複数の相手手札推定

手法をもとに対戦中にそれらを切り替えることを考える。この提案手法は、富岡 [7] が開発した UECda の 2020 年無差別級優勝クライアント Ninja に着想を得た。Ninja は、モンテカルロシミュレーションにおけるプレイアウト方法を選択バンディット法で動的に切り替えることで、相手のプレイスタイルに適応しようとした。しかし、最適解を求める方法を変えるよりは、対戦相手の提出手傾向に合わせた手札推定を目指す方がプレイアウト精度の向上に繋がり、より効果的なのではないかと考えた。

本研究は、Ninja の相手に応じた選択バンディット法を、相手手札推定方法に適用するものである。具体的には、近年の UECda 優勝プログラムのベースである Blauweregen をもとに、チートを含む 5 つの相手手札推定手法を切り替えるプレイヤを作成した。作成したプレイヤを、既存の UECda ライト級プログラムと対戦させることで、相手に応じてどのように最適な手札推定を発見するか、それによりプレイヤがどのように強化されるかを評価・検証する。

実験の結果、以下の知見を得た。まず、相手手札推定手法を切り替える手法は特別有効な推定手法が存在する場合には一定の有効性があると考えられる。他方、評価値の差が生じにくい対戦条件では、本手法によって最適な相手手札推定手法が有意に多く選択されるまでに時間がかかることもある。

本論文の構成は以下のとおりである。第 2 章では、本研究で用いるコンピュータ大貧民のルールを説明し、また本研究において使用する関連クライアントを紹介する。第 3 章では、相手手札推定の切り替えアルゴリズムについて説明し、本研究で用いる相手手札推定手法についての予備実験の結果も示す。第 4 章では、評価実験を行う。第一の評価実験では、評価対象プレイヤが 1 つの場合について、チートの有無によりどの相手手札推定手法がどれだけの割合で選ばれるかについて評価する。第二の評価実験では、評価対象プレイヤを複数用意し、試合開始時の階級の影響を評価値に含めた場合の選択バンディット法の動作について評価する。第三の評価実験では、対戦相手によって推定精度の変わる推定手法があった場合の挙動について評価する。これらの結果を受けて、第 5 章で本研究をまとめる。

第 2 章

コンピュータ大貧民

2.1 コンピュータ大貧民のルール

大貧民はトランプを用いた日本発祥のカードゲームの一つであり，手札を使い切る早さで順位と階級が決定する．数多くのローカルルールが存在するが，本研究では 2006 年から電気通信大学で開催されている UEC コンピュータ大貧民大会 (UECda) [3] の UEC 標準ルールを用いる．

UEC 標準ルールのうち重要なものを以下に示す．

試合の流れ 試合は 5 プレイヤで行われる．手札の配布・交換の後，各プレイヤが順に手札を提出する．スペード，ハート，ダイヤ，クラブのそれぞれ A～K とジョーカーの計 53 枚のカードを用いる．

階級 手札がなくなった順に順位が決定し，順位に応じて次試合での階級が割り振られる．階級は 1 位のプレイヤから順に大富豪，富豪，平民，貧民，大貧民であり，順に 5 点，4 点，3 点，2 点，1 点の得点が与えられる．

手札配布 各プレイヤにランダムに 10 枚か 11 枚の手札が配布される．

手札交換 手札配布後，大富豪と大貧民は 2 枚，富豪と貧民は 1 枚カードを交換する．大富豪と富豪は任意のカードを，大貧民と貧民は最も強いカードを渡す．

カード役 カード 1 枚のみを使用する単体役，同ランクのカードを複数使用する複数役，同マークで連続したランクのカードを 3 枚以上使用する階段役の 3 種類がある．

手札提出 ダイヤの 3 を持つプレイヤから試合を開始する．手番が回ってきたプレイヤは，

2.2 コンピュータ大貧民の既存クライアント

カードの提出かパスを行う。場に何もない状態であれば任意のカードを提出できる。場にカードが出ている場合は、場のカードと同じ役、同じ枚数で、より強い手のみ提出できる。

8切り 8を含んだ役を出した場合、場が流れる（場にカードがない状態となる）。

スペードの3 ジョーカーの単体役が提出されている場合、スペードの3を提出できる。その場合は場が流れる。

ジョーカー 単体役もしくは複数役において全てのカードの代用にすることができる。単体役として出した場合は、2より強い単体役となる。

革命 4枚以上の複数役もしくは5枚以上の階段役が出された場合、ジョーカー以外のカードの強さが逆転する。革命中に革命が起きた場合はカードの強さが元に戻る。

しばり 場と同じマークが出された場合、場が流れるまで同じマークのカードしか出すことができない。

席順変更 3試合に1回、各試合開始時にプレイヤーの席順がランダムに決定される。

2.2 コンピュータ大貧民の既存クライアント

UECda には無差別級とライト級の2種類の階級が存在する。

ライト級 基準クライアント（Nakanaka）の3倍以下の実行時間の階級。制限が厳しいため、ヒューリスティック戦略が主流となっている。

無差別級 初代無差別級優勝クライアント（jnishino）と同程度、またはそれ以下の実行時間の階級。比較的制限が小さく、モンテカルロ法が主流となっている。

本研究に関連する既存クライアントを以下に示す。

2.2 コンピュータ大貧民の既存クライアント

2.2.1 Blauwereggen

大渡が開発した Blauwereggen [4] は 2017 年無差別級優勝クライアントである。Blauwereggen では、方策勾配法により調整した方策関数を使用してモンテカルロシミュレーションを行う。その際、採択棄却法 (REJECTION) により相手手札推定を行っている。デフォルト設定では設定した計算時間を使い切るまで、着手候補数や計算時間に応じて 5000 回 (手札交換の際は 10000 回) を上限にシミュレーションが行われる。本研究では、Blauwereggen の実装に含まれている 4 つの相手手札推定に、チートを加えた 5 つを用いる。

2.2.2 Ninja

富岡が開発した Ninja [7] は 2020 年無差別級優勝クライアントである。Ninja は、Blauwereggen をベースとし、そのモンテカルロ法部分を改良している。具体的には、モンテカルロ法で用いるバンディットアルゴリズムとして UCB-root, UCB1-Tuned, UCB2, KLUCB, UCB-V の 5 つを事前に用意し、それらのうち採用するアルゴリズムを選択バンディット法により決定する。

2.2.3 switch2

switch2 は 2022 年 UECda ライト級優勝クライアントである。switch2 では複数の戦略を実装しており、最初の 1500 試合ではそれらを均等に使用するが、その後 1500 試合では勝率の高かった戦略を優先使用し、以降は勝率が上位の戦略のみを使用する。

2.2.4 PoliXLite

PoliXLite は 2020 年 UECda ライト級優勝クライアントである。PoliXLite も Blauwereggen をベースとしているが、モンテカルロシミュレーションを行わずに方策関数の結果をそのまま用いて手を選択している。

2.3 相手手札推定手法

相手手札推定手法として、Blauweregen の実装に含まれている以下の 4 種類の方法と、すべての相手手札を正しく埋めるチート (CHEAT) を利用する。

RANDOM 自分の手札以外を完全にランダムに分配する。

SBJINFO 交換等で判明している確実な情報のみ埋め、残りはランダムに分配する。

BIAS 逆関数法を用いて、重みづけに基づいたカード分配を行う。分配の際は各プレイヤーの階級から所持確率を計算する。

REJECTION 採択棄却法を用いて複数のカード配置候補を生成し、方策関数で算出した尤度が最大の配置を選択する。

CHEAT サーバから送られた相手手札情報を用いて正確に埋める。

Blauweregen で採用されている推定方法は **REJECTION** だが、ゲーム後半になると計算量が組み合わせ爆発を起こしてしまう。そのため、手札配置の作成においてはカード配置候補の生成回数に上限を設け、失敗回数が一定回数を超えると **BIAS** による計算に切り替わる。また、推定は使用する手札配置が一定個数に達するまで何度も行われる。

REJECTION については以下のような処理が行われる。

1. 所持が確定しているカード情報を埋める。
2. カード交換後のカード配置に近い分布となる手札配置を一定個数作成する。
3. 各配置について、当試合の試合進行が生まれるような尤度を計算する。
4. 尤度最大の手札配置セットを相手手札候補として採用する。

また、**BIAS** については以下のような処理が行われる。

1. 所持が確定しているカード情報を埋める。
2. 未確定カードを 1 枚取り出し (強い順)、そのカードの累積確率分布を計算する。
3. 一様乱数を生成し、累積分布から該当するプレイヤーにカードを割り当てる。

2.3 相手手札推定手法

4. (2)～(3) を繰り返し、所持未確定カードがなくなれば終了.

実験に用いる各種手札推定手法の強さを確認する予備実験を行った. 実験には推定手法を固定し、シミュレーション回数を 5000 回に設定した改造 Blauweregen を用い、以下の組み合わせで 10000 試合からなる対戦を 10 回行った.

予備実験 1 switch2 × 4, 改造 Blauweregen × 1

予備実験 2 PoliXLite × 4, 改造 Blauweregen × 1

予備実験 3 switch2 × 2, PoliXLite × 2, 改造 Blauweregen × 1

各推定手法を用いた改造 Blauweregen の獲得得点を表 2.1 に、各推定手法の推定精度を表 2.2 に示す. 相手手札推定手法を固定したプレイヤはおよそ, $\text{RANDOM} \approx \text{SBJINFO} \approx \text{BIAS} < \text{REJECTION} < \text{CHEAT}$ の順に並んでいる. REJECTION は, Blauweregen に実装された他 3 つの推定方法と 1000 点以上離れている. また, CHEAT は対 switch2 において 4000 点以上, 対 PoliXLite において 6500 点以上と, 大きく REJECTION を上回っている.

Blauweregen にデフォルトで搭載されている中で推定精度の最も高い REJECTION でも 36～37%の精度でしか相手手札を推定できないことから、大貧民の相手手札推定精度はあまり高くないことが読み取れる. RANDOM と SBJINFO, BIAS と REJECTION ではそれぞれ 3%程度推定精度の差がある. 前者の得点の差は標準偏差の範囲を出ないが、後者は明確な得点差が表れている. これは、完全一致でないカードであっても近しい強さのカードを選択できているなど、実際の手札配置に近い推定が実現しているためだと推測できる.

2.3 相手手札推定手法

表 2.1 各推定手法を用いた改造 Blauweregen の獲得得点

対戦相手	switch2	PoliXLite	2 体ずつ
	平均 ±標準偏差	平均 ±標準偏差	平均 ±標準偏差
RANDOM	35563.2± 272.3	34964.2± 311.9	34786.4± 159.9
SBJINFO	35629.5± 207.8	35221.7± 244.8	34942.5± 233.1
BIAS	35869.0± 195.1	35520.6± 281.6	35170.6± 246.8
REJECTION	37079.1± 205.1	36832.2± 225.2	36480.3± 233.1
CHEAT	41234.8± 167.1	43370.9± 242.5	42123.1± 159.9

表 2.2 各推定手法の推定精度

対戦相手	switch2	PoliXLite	2 体ずつ
RANDOM	28.04%	28.49%	28.28%
SBJINFO	31.16%	31.22%	31.21%
BIAS	33.24%	33.18%	33.16%
REJECTION	36.26%	36.74%	36.67%
CHEAT	100.00%	100.00%	100.00%

第 3 章

提案手法

本研究で提案するアルゴリズムは，事前に用意した複数の相手手札推定手法を用いて，モンテカルロ法の相手手札推定時にそれらを切り替えるというものである．相手手札推定方法の切り替えは試合ごとに行い，各試合で選択する相手手札推定方法は UCB-V により決定する（Ninja の選択バンディット法と同じ）．なお，特に断りのない限り，評価値の更新には得点を用いる．本研究で提案する相手手札推定手法の切り替え方法の概念図を，図 3.1 に示す．

提案アルゴリズムは以下の手順で動作する．

1. バンディットアルゴリズム UCB-V に基づき，その試合で用いる相手手札推定手法を決定する．
2. 選択した相手手札推定手法で 1 試合対戦する．
3. 試合の結果から，相手手札推定手法の評価値を更新する．
4. 指定の試合数 (1)～(3) を繰り返す．

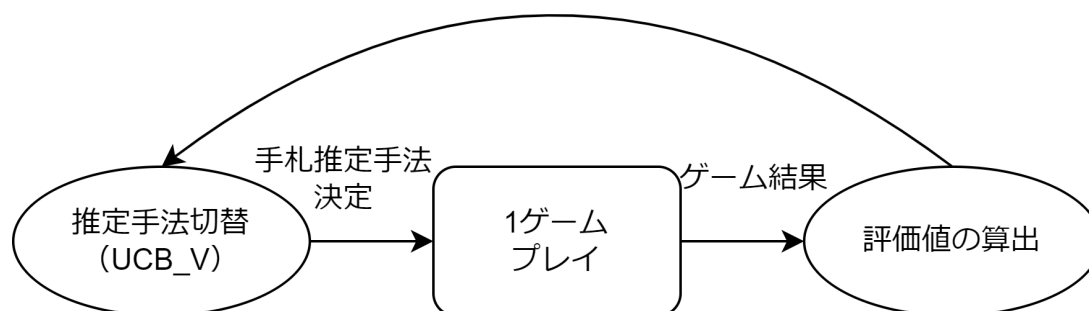


図 3.1 相手手札推定アルゴリズムの切り替え方法

UCB-V (Upper Confidence Bound with Variance) は, 報酬の分散を考慮したバンディットアルゴリズムであり, 不確実性が高い環境での探索に適している. UCB-V では, 以下の基準に基づきアームを選択する.

$$\arg \max_{i \in A} \left(\hat{x}_i + \sqrt{\frac{2\hat{\sigma}_i E(s)}{n_i}} + c \frac{3bE(s)}{n_i} \right)$$

ここで, 探索のスケールを調整する関数 $E(t)$ は以下のように定義される.

$$E(t) = \zeta \log(t)$$

- $\hat{x}_i$: アーム i の報酬平均
- $\hat{\sigma}_i$: アーム i の報酬分散
- ζ, c : 正の定数 ($\zeta > 0$)
- b : 定数

第 4 章

実験

4.1 評価対象プレイヤーが 1 つの場合の評価

Blauweregen をベースに，相手手札推定手法を動的に切り替える提案手法を実装したプログラム Kunoichi を用いて評価実験を行う．評価実験では相手手札推定手法が収束するか，推定手法の変更により強くなるかを確認する．具体的には，CHEAT を含む場合には CHEAT が，CHEAT を含まない場合には REJECTION が最も高頻度に出ることが望む結果である．

実験には，提案手法によるクライアントとして，相手手札推定手法に CHEAT を含まない Kunoichi, 相手手札推定手法 CHEAT を含む cheat_Kunoichi の 2 つを用いる．それらの対戦相手には，switch2 と PoliX Lite を用いる．Kunoichi および cheat_Kunoichi のシミュレーション回数を 5000 回に設定し，以下の組み合わせで 10000 試合からなる対戦を 10 回行った．

実験 1-S-C switch2 × 4, cheat_Kunoichi

実験 1-P-C PoliX Lite × 4, cheat_Kunoichi

実験 1-M-C switch2 × 2, PoliX Lite × 2, cheat_Kunoichi

実験 1-S-K switch2 × 4, Kunoichi

実験 1-P-K PoliX Lite × 4, Kunoichi

実験 1-M-K switch2 × 2, PoliX Lite × 2, Kunoichi

各実験における相手手札推定方法の累積選択割合の推移を図 4.1～4.6 にそれぞれ示す．表

4.1 評価対象プレイヤーが1つの場合の評価

4.1 に各実験における評価プレイヤー (cheat_Kunoichi, Kunoichi) の獲得得点の平均を示す。また、最終的な相手手札推定方法の選択回数、および、選択された試合での獲得得点の平均を、表 4.2 と表 4.3 に示す。

Kunoichi の得点は Blauweregen よりは下がっているが、その差は最大で対 PoliX Lite の 571.3 点と十分に小さい。cheat_Kunoichi の CHEAT のみとの得点差もいずれの対戦相手でも 500 点未満であり、こちらもそれほど大きな差はみられない。

CHEAT ありの対戦では、どの対戦相手であっても全体で 90%以上の確率で CHEAT が選ばれている。一方で、CHEAT なしの対戦では 70%前後と CHEAT ありと比較して低い確率に留まった。これは実験 1-P-K や実験 1-M-K において BIAS が多く選ばれていたことが影響していると推測できる。

全体として、CHEAT ありでは 1000 から 1500 試合、CHEAT なしでも 6000~8000 試合程度で直近 100 試合で最適な相手手札推定手法を選択した割合が 9 割を超えている。このことから、特に有用な推定手法があった場合にはきちんとその推定方法に収束することが読み取れる。

4.1 評価対象プレイヤーが1つの場合の評価

表 4.1 実験 1 における評価プレイヤーと Blauweregen の獲得得点平均

対戦相手	switch2	PoliXLite	2 体ずつ
	平均 ±標準偏差	平均 ±標準偏差	平均 ±標準偏差
cheat_Kunoichi	40792.3± 207.1	42989.7± 126.0	41498.7± 186.3
Kunoichi	36586.5± 239.1	36260.9± 338.3	36124.8± 365.6
Blauweregen	37079.1± 205.1	36832.2± 225.2	36480.3± 233.1

表 4.2 実験 1-S-C, 実験 1-P-C, 実験 1-M-C (CHEAT あり) において, 各相手手札推定手法が選択された回数と, その試合での獲得得点の平均 (括弧内)

対戦相手	RANDOM	SBJINFO	BIAS	REJECTION	CHEAT
switch2	201.4	227.7	269.8	253.8	9047.3
	(3.5487)	(3.6004)	(3.6820)	(3.6690)	(4.1264)
PoliXLite	167.6	167.3	150.7	152.2	9362.2
	(3.6611)	(3.6623)	(3.5839)	(3.5926)	(4.3448)
2 体ずつ	161.1	160.5	188.2	207.5	9282.7
	(3.4786)	(3.4798)	(3.5765)	(3.6410)	(4.1961)

表 4.3 実験 1-S-K, 実験 1-P-K, 実験 1-M-K (CHEAT なし) において, 各相手手札推定手法が選択された回数と, その試合での獲得得点の平均 (括弧内)

対戦相手	RANDOM	SBJINFO	BIAS	REJECTION	CHEAT
switch2	851.1	907.7	874.4	7366.8	—
	(3.5560)	(3.5699)	(3.5581)	(3.6934)	
PoliXLite	817.2	779.2	1100.9	7302.7	—
	(3.4969)	(3.5069)	(3.5278)	(3.6681)	
2 体ずつ	617.7	1225.8	1375.6	6780.9	—
	(3.4586)	(3.5152)	(3.5398)	(3.6588)	

4.1 評価対象プレイヤーが1つの場合の評価

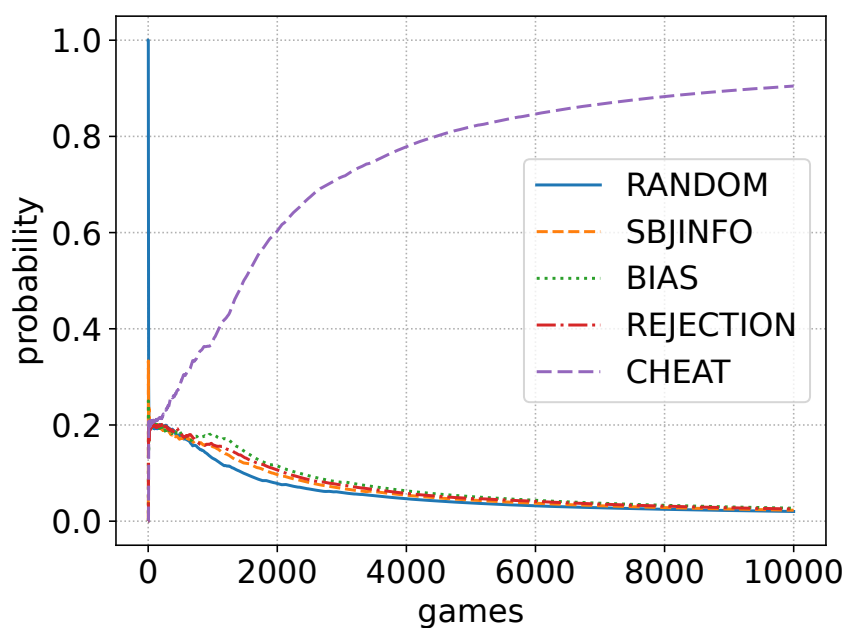


図 4.1 実験 1-S-C (switch2, CHEAT あり) における推定方法累積選択割合の推移

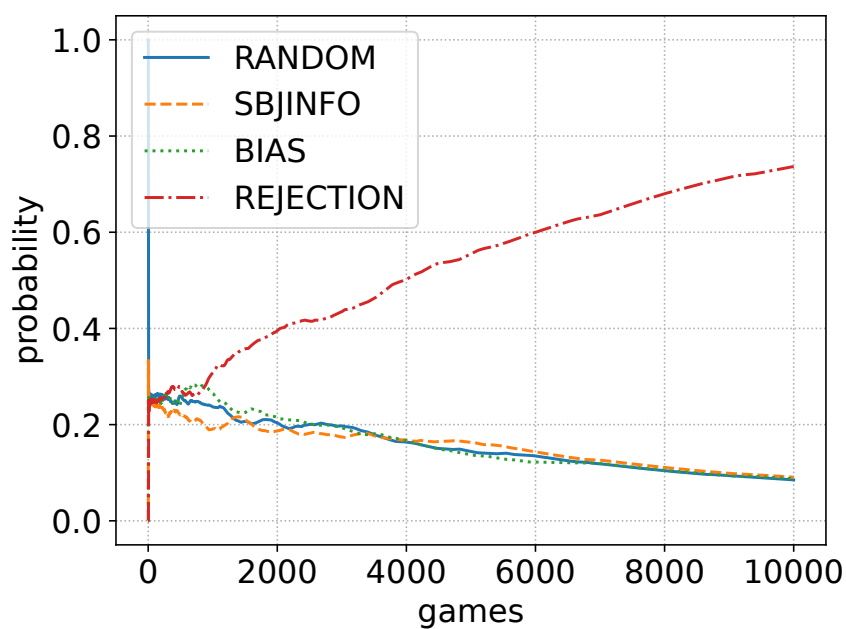


図 4.2 実験 1-S-K (switch2, CHEAT なし) における推定方法累積選択割合の推移

4.1 評価対象プレイヤーが1つの場合の評価

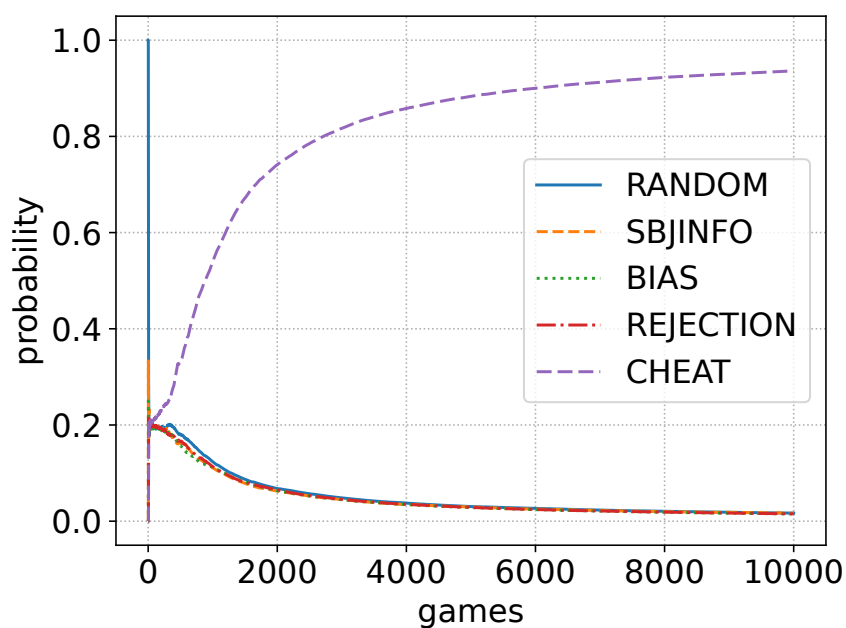


図 4.3 実験 1-P-C (PoliX Lite, CHEAT あり) における推定方法累積選択割合の推移

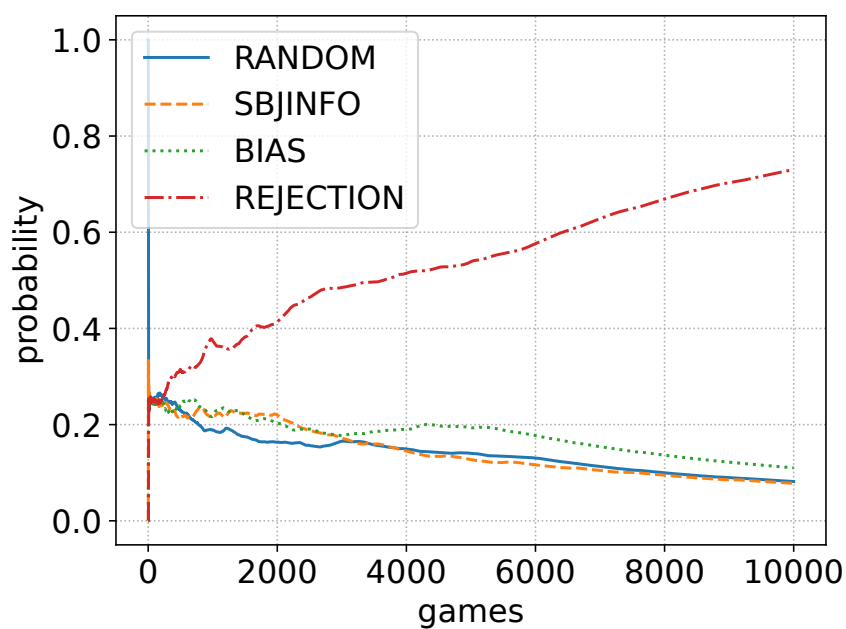


図 4.4 実験 1-P-K (PoliX Lite, CHEAT なし) における推定方法累積選択割合の推移

4.1 評価対象プレイヤーが1つの場合の評価

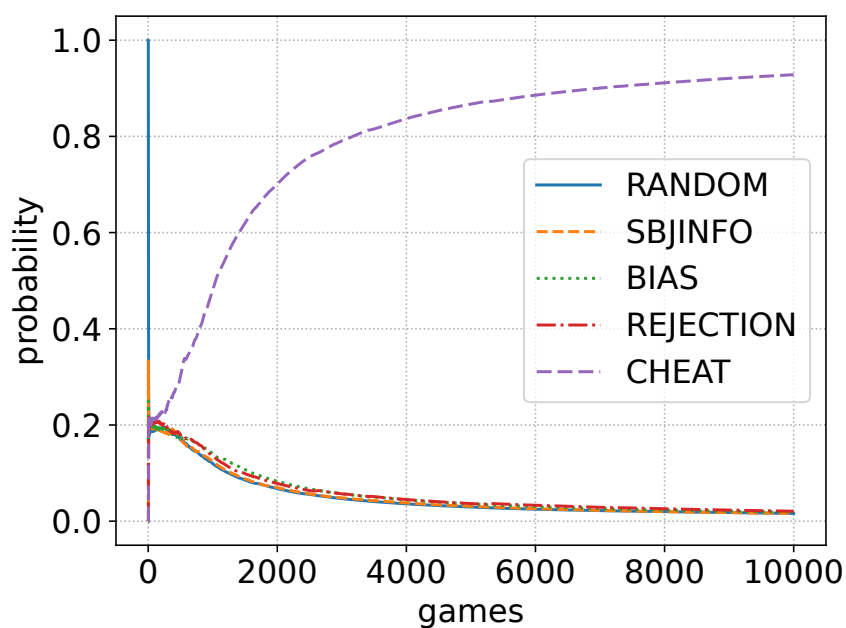


図 4.5 実験 1-M-C (2 種類のプレイヤー, CHEAT あり) における推定方法累積選択割合の推移

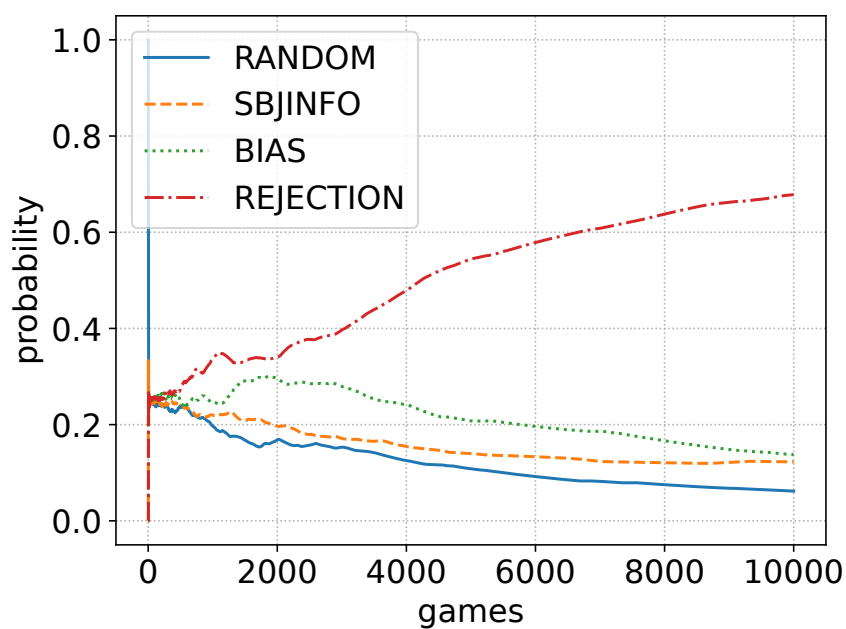


図 4.6 実験 1-M-K (2 種類のプレイヤー, CHEAT なし) における推定方法累積選択割合の推移

4.2 試合開始時の階級を考慮した評価値を用いる効果の評価

評価対象プレイヤーが1つの実験では、対戦相手は UECda ライト級のクライアントを対戦相手として利用した。しかし、自身と強さが同程度の相手と対戦する際に強い推定手法に収束しなければ実際に対戦するうえで性能を発揮することは難しい。この実験では、自身と同程度の性能の相手でも収束することが確認できるか、その際の評価値の設定方法が適切であるかを調査する。

Ninja のバンディットアルゴリズムの切り替えに用いる評価値は、得点のみから算出されるものであった。大貧民は前の試合の勝者（大富豪、富豪）が勝ちやすく、敗者（大貧民、貧民）が負けやすいゲームである。このことを考慮して、以下のように評価値の与え方を変更することを考える（階級考慮あり，WR）。

- 前の試合の結果が大貧民または貧民であるとき、評価値を獲得点数 +1 とする。
- 前の試合の結果が平民であるとき、評価値を獲得点数とする。
- 前の試合の結果が大富豪または富豪であるとき、評価値を獲得点数 -1 とする。

なお、階級考慮を適用するのは評価対象プレイヤーのみとする。

実験にはシミュレーション回数を 5000 回に固定した Kunoichi と UECda の 2022 年ライト級優勝クライアントである switch2 を用い、以下の組み合わせで 10000 試合からなる対戦を 10 回行った。

実験 2-PT-K4 評価値は得点のみ，switch2，Kunoichi×4

実験 2-PT-K3 評価値は得点のみ，switch2×2，Kunoichi×3

実験 2-WR-K4 評価値に階級考慮，switch2，Kunoichi×4

実験 2-WR-K3 評価値に階級考慮，switch2×2，Kunoichi×3

評価プレイヤー（プレイヤー 4 とする）の獲得得点の合計を表 4.4 に示す。また、各相手手札推定方法の累積選択割合の推移を図 4.7～4.10 に示す。

4.2 試合開始時の階級を考慮した評価値を用いる効果の評価

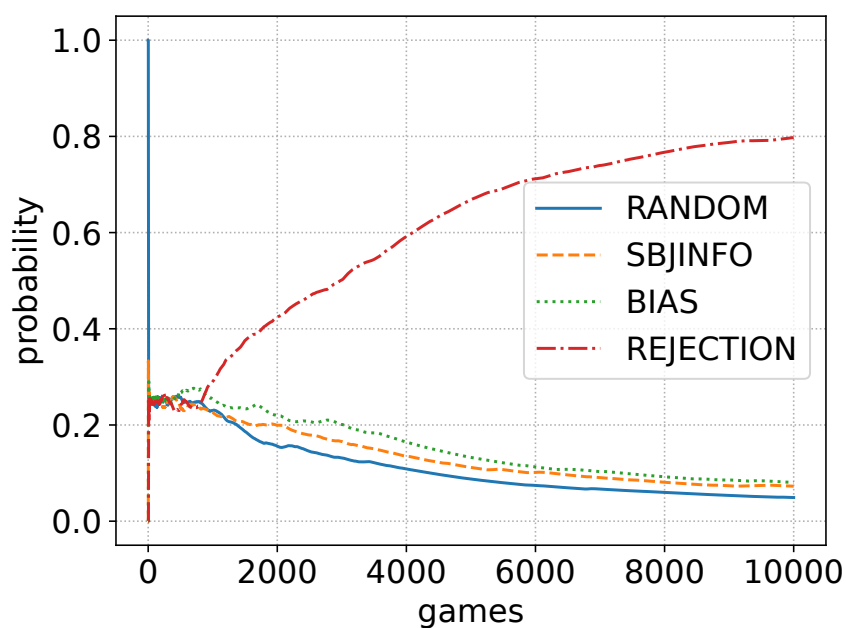


図 4.7 実験 2-PT-K4（得点のみ，Kunoichi×4）における推定方法累積選択割合の推移

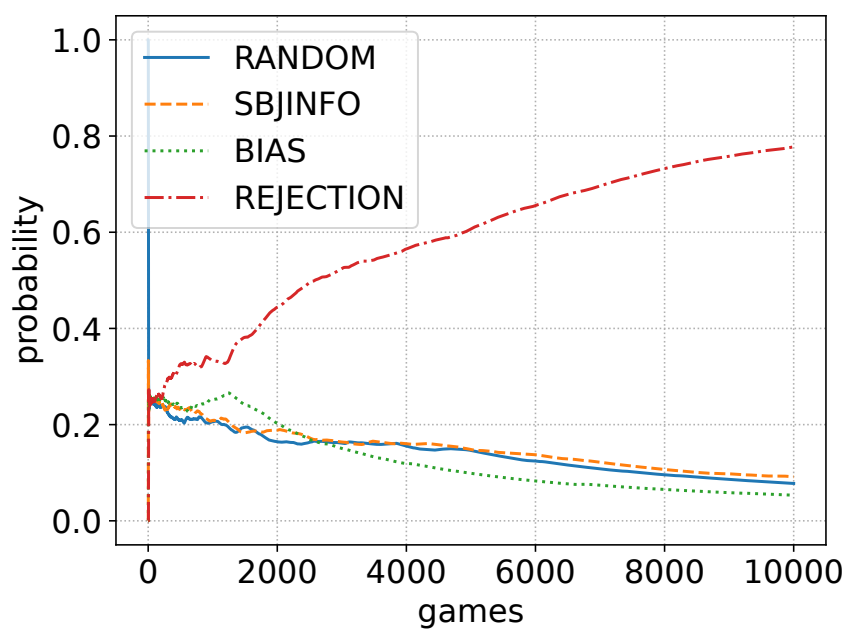


図 4.8 実験 2-WR-K4（階級考慮，Kunoichi×4）における推定方法累積選択割合の推移

4.2 試合開始時の階級を考慮した評価値を用いる効果の評価

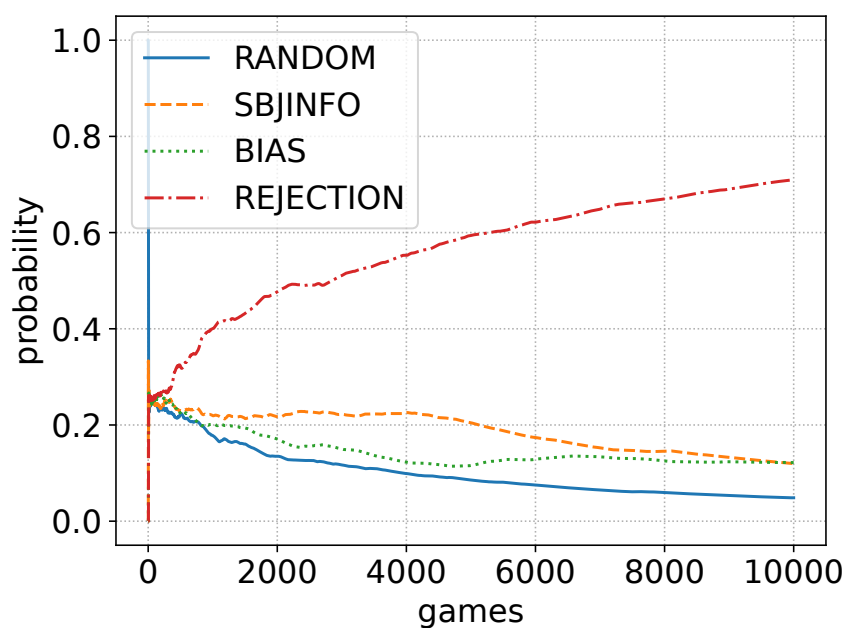


図 4.9 実験 2-PT-K3（得点のみ，Kunoichi×3）における推定方法累積選択割合の推移

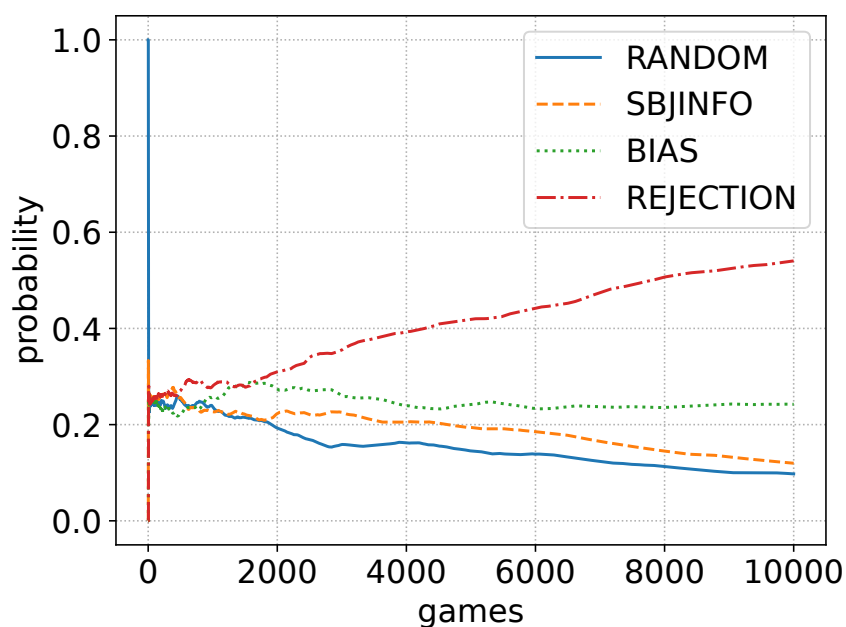


図 4.10 実験 2-WR-K3（階級考慮，Kunoichi×3）における推定方法累積選択割合の推移

4.3 対戦相手によって推定精度が大きく異なる場合の評価

表 4.4 実験 2 における評価プレイヤーの獲得得点合計

対戦相手	Kunoichi × 4	Kunoichi × 3
	平均 ±標準偏差	平均 ±標準偏差
評価値：得点のみ	31805.3± 325.4	33195.0± 412.4
評価値：階級考慮	31469.2± 381.0	32989.2± 464.3

評価プレイヤーの獲得得点は、得点のみで評価値を設定した方が高い結果となった。累積選択割合の推移も WR-K3 では半数の 5 つが REJECTION 以外へと収束しようとする様子を見せ、その中でも最も弱い RANDOM に収束した試合が 2 つ存在した。全体の REJECTION 選択率も 6 割を切るなど著しく低く、直近 100 試合で最適な相手手札推定手法を選択した割合も 7 割に 5586 試合でようやく到達したものの、最後まで 8 割にすら届かなかった。その他の条件では PT-K4, PT-K3 では 4500 試合、WR-K4 では 3000 試合程度で 9 割に届き、自分より弱いプレイヤー相手の時よりも明確な差が現れた。

これらの結果から、自分と近い強さのプレイヤー相手の方が収束が早いこと、階級はこの方法で考慮したとしてもあまり有効には働かないことなどが読み取れた。

4.3 対戦相手によって推定精度が大きく異なる場合の評価

これまでの実験はどの相手にも最も強い推定手法が一致していた。実戦では対戦相手はそれぞれ違うプレイヤーになり、最適な推定手法もプレイヤー間で異なるはずである。この実験では、対戦相手によって推定精度に差が出るような推定方法を作成し、その挙動について明らかにする。

この実験では、対戦相手として switch2 と PoliX Lite を 2 つずつ対戦相手とし、Kunoichi_cheat の CHEAT の挙動を以下のように変更した。

- switch2 に対しては、そのプレイヤーの手札のうち $n\%$ を弱い順で推定した後、残りを RANDOM で埋める。

4.3 対戦相手によって推定精度が大きく異なる場合の評価

表 4.5 実験 3 における評価プレイヤーの獲得得点平均と CHEAT の推定精度

	平均 ±標準偏差	CHEAT の推定精度
50%	36084.3± 385.6	48.38%
60%	35943.4± 281.0	52.50%
70%	35973.7± 465.9	56.84%
80%	36235.2± 256.4	62.54%
90%	36541.4± 347.0	68.63%
100%	38363.4± 239.7	78.37%

- PoliX Lite に対しては、RANDOM で手札を推定する。

n を 50, 60, 70, 80, 90, 100 に設定し、10000 試合からなる対戦を 10 回行った。

評価プレイヤーの獲得得点平均と推定精度を表 4.5 に示す。また、各相手手札推定方法の累積選択割合の推移を図 4.11～4.16 に示す。

REJECTION は 37%程度の推定精度しか持たないはずだが、CHEAT の推定精度が 60%を超える $n = 80$ になってようやく同程度選ばれるようになった。このことから、現状の方法では単に特定のプレイヤーのカードが確実に推定できるのみでは不十分であったといえる。

$n = 50$ の段階では基本的には REJECTION に収束するが、10 セット中 3 セットは CHEAT への収束がみられた。 $n = 80$ になるとほぼ同程度に選ばれるようになり、 $n = 90$, $n = 100$ では CHEAT への収束がみられた。

4.3 対戦相手によって推定精度が大きく異なる場合の評価

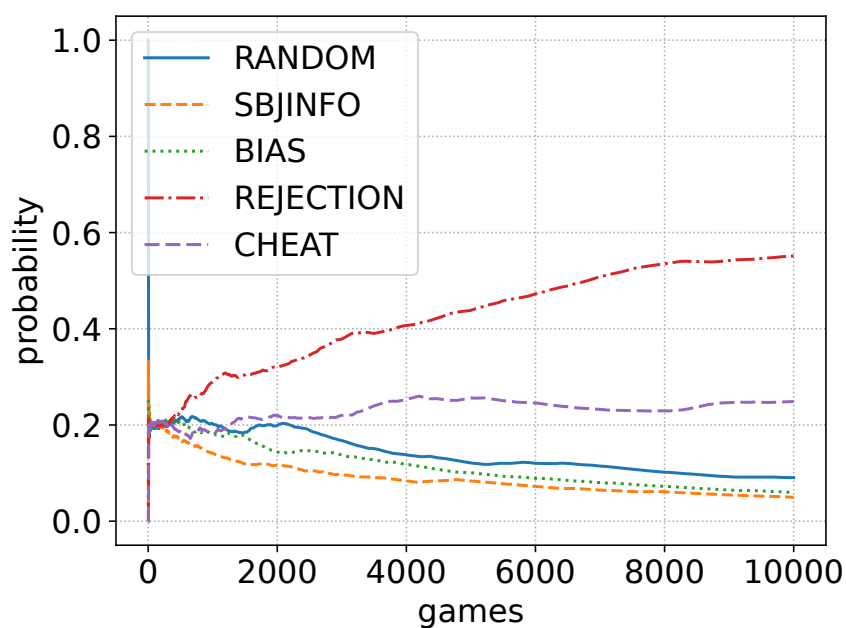


図 4.11 $n = 50$ における各推定方法の累積選択割合の推移

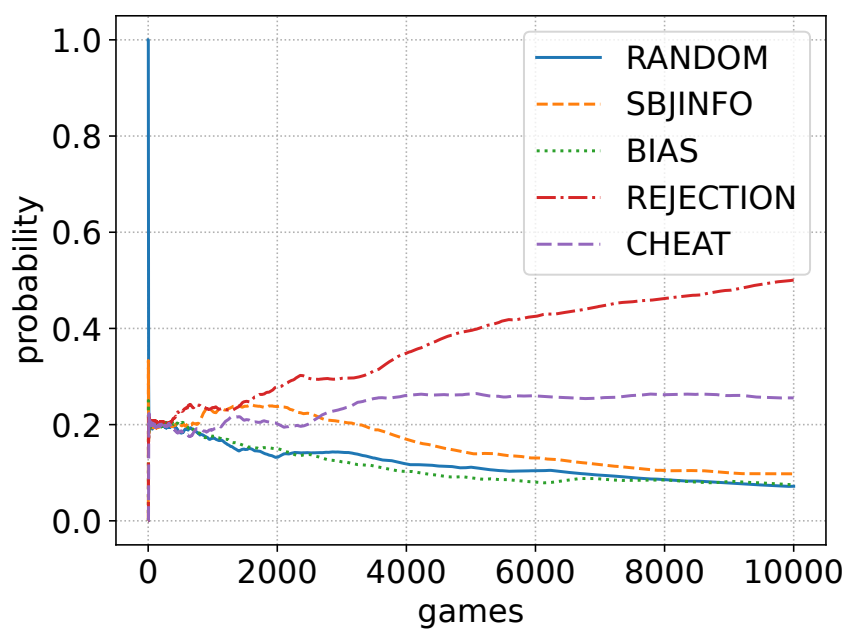


図 4.12 $n = 60$ における各推定方法の累積選択割合の推移

4.3 対戦相手によって推定精度が大きく異なる場合の評価

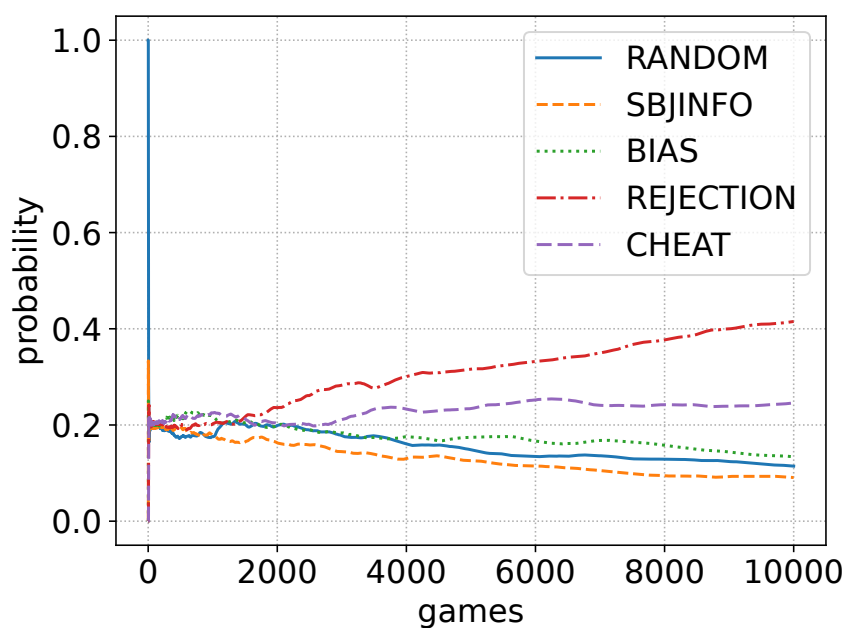


図 4.13 $n = 70$ における各推定方法の累積選択割合の推移

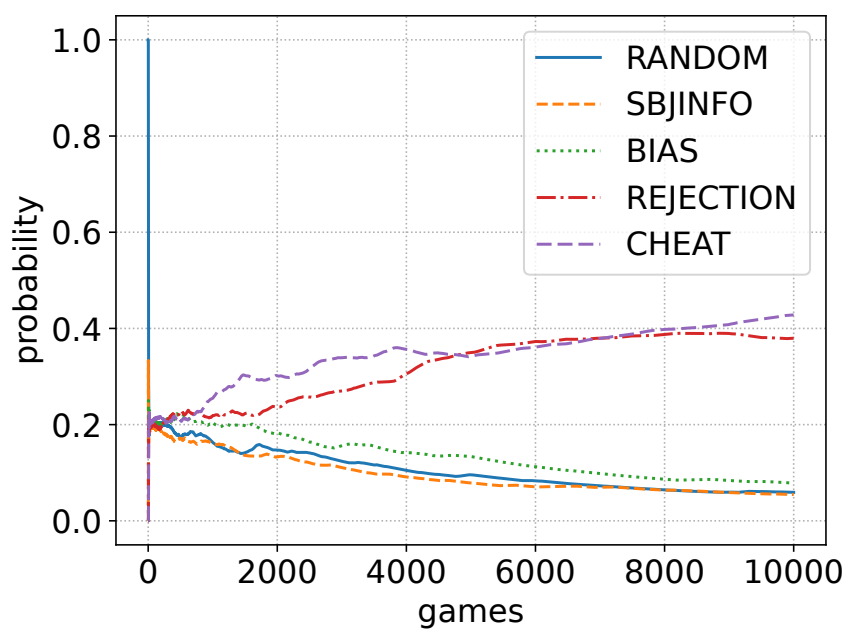


図 4.14 $n = 80$ における各推定方法の累積選択割合の推移

4.3 対戦相手によって推定精度が大きく異なる場合の評価

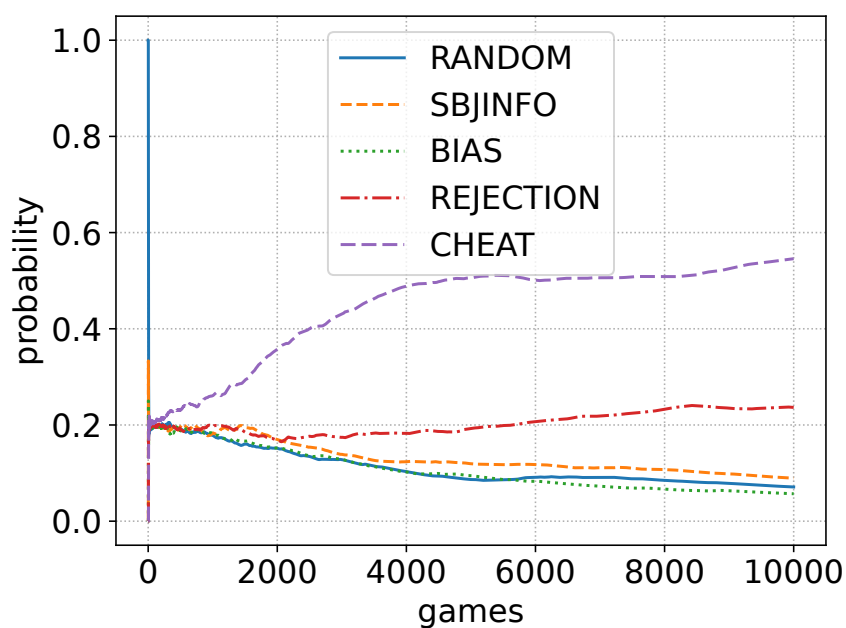


図 4.15 $n = 90$ における各推定方法の累積選択割合の推移

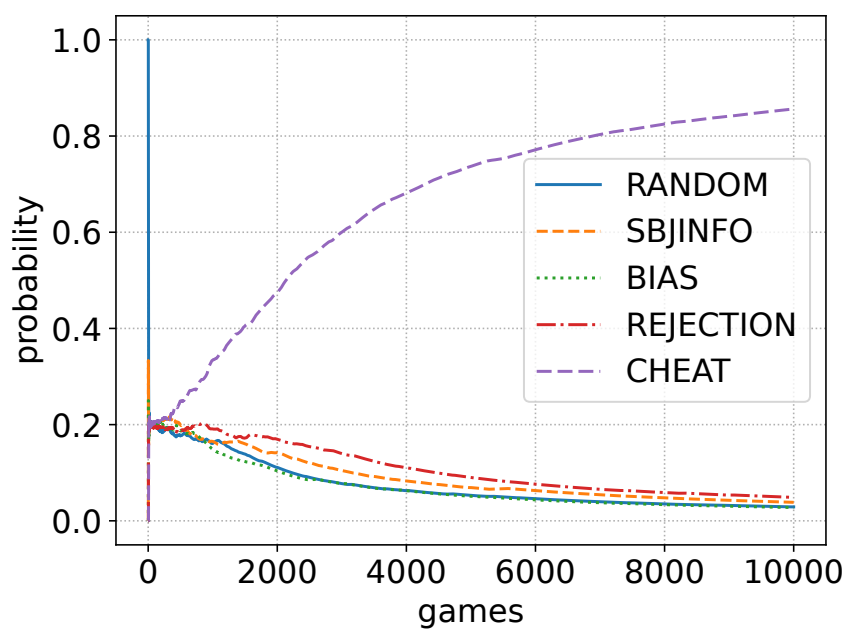


図 4.16 $n = 100$ における各推定方法の累積選択割合の推移

第 5 章

まとめ

本研究の目的は、相手手札推定方法を動的に切り替える手法の提案ならびにその有効性の検証である。実験の結果、条件が良ければ 1000 試合程度、そうでない場合も 8000 試合程度で過去 100 試合で 9 割以上最適な推定方法に収束することが示された。

CHEAT ありの対戦では必ず、CHEAT が頻繁に選ばれていたため、相手手札推定手法を切り替える手法は特別有効な推定手法が存在する場合には一定の有効性があると考えられる。

実験 2 において、ライト級クライアントとの対戦よりも、無差別級クライアント同士での対戦の方が最適な推定方法に早く収束したことは、興味深い結果である。一方で、前試合の階級を考慮した評価値を用いた実験では、得意な相手には僅かにしか強くならず、そうでない相手に対しては最適でない相手手札推定手法が多く選ばれる結果となった。大貧民は前の試合の結果による手札交換が大きく影響するゲームではあるが、最初に配られる手札の強弱の影響の方が大きい。試合開始時の手札の強弱を反映した評価値算出等により精度を上げられる可能性が考えられる。

今回の研究ではすべてのプレイヤーに対して同じ推定手法を利用した。しかし、実験 3 では推定精度が 60%になるまで REJECTION を超えられないなど、現状の方法では特定プレイヤーの手札推定だけができたとしても不十分であったといえる。対戦相手それぞれに別の推定方法を選べるようにすることが今後の重要な課題である。

謝辞

本研究を進めるにあたり，熱心なご指導を頂いた担当教員の松崎公紀教授，高田喜朗教授，竹内聖悟講師に御礼申し上げます。

Blauweregen の作者である大渡勝己さまには，推定アルゴリズムの手法についての大変有益なご説明を頂きました。心より感謝いたします。

参考文献

- [1] D. Silver, J. Schrittwieser, K. Simonyan, I. Antonoglou, A. Huang, A. Guez, T. Hubert, L. Baker, M. Lai, A. Bolton, Y. Chen, T. Lillicrap, F. Hui, L. Sifre, G. van den Driessche, T. Graepel and D. Hassabis: Mastering the game of Go without human knowledge, *Nature*, **550**:354–359, 2017.
- [2] D. Silver, T. Hubert, J. Schrittwieser, I. Antonoglou, M. Lai, A. Guez, M. Lanctot, L. Sifre, D. Kumaran, T. Graepel, T. Lillicrap, K. Simonyan and D. Hassabis: A general reinforcement learning algorithm that masters chess, shogi, and Go through self-play, *Science*, **362**(6419):1140–1144, 2018.
- [3] UECda-2023: <https://flute.u-shizuoka-ken.ac.jp/daihinmin/2023/> (2023) .
- [4] 大渡勝己, 田中哲朗: 方策勾配を用いた教師有り学習によるコンピュータ大貧民の方策関数の学習とモンテカルロシミュレーションへの利用, 情報処理学会研究報告, 2016-GI-35, No. 10, pp. 1–8 (2016).
- [5] 神田直樹, 伊藤毅志: コンピュータ大貧民における LSTM を用いた手札推定, 情報処理学会研究報告ゲーム情報学 (GI), 2018-GI-39(8), pp. 1–8, 2018.
- [6] 須藤郁弥, 成澤和志, 篠原歩: UEC コンピュータ大貧民大会向けクライアント「snow1」の開発, 第 2 回 UEC コンピュータ大貧民シンポジウム, 2010.
- [7] 富岡聖, 大久保誠也, 湯瀬裕昭, 武藤申明: コンピュータ大貧民における各種モンテカルロ法の検討, 情報処理学会研究報告, 2021-GI-45, No.12, pp. 1–8 (2021).
- [8] 中山友里歌, 松崎公紀: コンピュータ大貧民における相手手札推定を切り替えるプレイヤの検討, ゲームプログラミングワークショップ 2024 論文集, pp. 173–178 (2024).
- [9] 中山友里歌, 植田桂広, 鴨川翔太, 松崎公紀: コンピュータ大貧民における手札推定の有効性についての再評価と考察, 情報処理学会研究報告, 2022-GI-48, No. 6, pp. 1–7

参考文献

(2022).

- [10] 西野順二, 西野哲朗: 大貧民における相手手札推定, 情報処理学会研究報告数理モデル化と問題解決 (MPS), 2011-MPS-85(9), pp. 1–6, 2011.
- [11] 濱田恭輔, 鈴木海友, 松澤智史: 大貧民エージェントの強化に向けた手札推定, 2023-GI-49, No. 5, pp. 1–7 (2023).
- [12] 柳澤佑介, 松崎公紀: 大貧民における出現頻度と提出手役履歴を用いた相手手札推定, 情報処理学会研究報告ゲーム情報学 (GI), 2015-GI-33(9), pp. 1–6, 2015.