Vol.2021-GI-45 No.13
2021/3/5

コンピュータ大貧民における戦略の定式化に関する研究

川崎 伊織[1]　大久保 誠也[2]　若月 光夫[1]　西野 哲朗[1]

概要：近年, コンピュータ大貧民の研究において機械学習を使用したプログラムが多く使用されている. これらのプログラムの多くは事前学習されたパラメータを持っており, 単体ではプログラムがどのように動作するのか理解が困難である. そこで, 大貧民プログラムの挙動を人間が理解しやすくするために, 学習パラメータに一定の規則があるかの検証を行い, 発見した規則を数式として表して定式化する手法を提案する. この定式化の過程として, まず, 事前学習されたパラメータをそれぞれの値が適用される状況によって分類し, グラフ上にプロットする. 次に, 同じ状況における評価値の増減を近似直線で表すことによって数式化する. このような数式に基づいて動作するように変更したプログラムの強さが元の学習パラメータを用いたプログラムから大きく乖離していないか, 特定の大貧民プログラムに適用して実験を行った. その結果, 提案手法によって変更したプログラムと元の学習パラメータによるプログラムには, 強さにおいて大きな変化はなかった. したがって, 学習パラメータから規則を見つけ数式として表現できたといえる.

1. はじめに

ゲーム情報学では, 完全情報ゲームと不完全情報ゲームに分けられて研究が行われている. 完全情報ゲームではAlphaGo[1] など人間に勝利するほどの成果が出ている. 一方で, 不完全情報ゲームでは, 人狼ゲームで人狼知能 [6] が企画され, 麻雀で Microsoft の麻雀 AI Suphx[7] がオンライン麻雀サイト天鳳で十段になるなど盛んである. トランプゲームの一つである大貧民では, コンピュータ同士で大貧民を対戦させる UEC コンピュータ大貧民大会 (以降, UECda と呼ぶ) が電気通信大学で開催されており, 大会では数千試合における順位の合計得点によって対戦プログラムの強さを競い合っている [2]. 同大会では, ヒューリスティックな手法を使ったプログラムが中心のライト級と, 機械学習を使ったプログラムが中心の無差別級の 2 部門に分かれている.

ライト級のプログラムは田頭ら [3] の kou2 をはじめとしたヒューリスティックな戦略を実装しているものが多い. ライト級プログラムを改良する際は, 新たな戦略を実装することが主になっており, 比較的容易に改良することができる. しかし, 強さとしては無差別級プログラムには遠く及ばないことが多い. 一方, 無差別級のプログラムは事前に学習されたパラメータや, モンテカルロ法などを用いた実装をしており, 非常に強いが単体では実際にどのように動作をするのか理解が困難である. また, モンテカルロ法は乱数を用いてシミュレーションを複数回行うことで近似的に解を求めるアルゴリズムのことであり, このシミュレーションで選ばれる解をシミュレーション前に理解することは難しい.

本研究の目的は, 事前に学習されたパラメータから一定の規則を見つけ, 数式として表す手法の確立である. 本研究では大貧民のルールやプレイヤの手札のカード, 提出しようとしている着手のカードに特に注目する. また, 数式化された規則が元の学習パラメータから大きく乖離していないかを確認するために計算機実験を行い, 所望の結果を得た.

2. コンピュータ大貧民

2.1 コンピュータ大貧民とは

コンピュータ大貧民とは, トランプゲームである大貧民を計算機上で行うゲームである. コンピュータ大貧民の研究では主に UECda の標準ルールが使用されている. 本項ではコンピュータ大貧民の具体的なルールやゲームの流れ, UECda で採用されているローカルルールなどを記述する.

- トランプの各スートの A から K までの 52 枚に, ジョーカー 1 枚を加えた計 53 枚でゲームを行う.
- 3 が一番弱く, 3, 4, 5, 6, 7, 8, 9, 10, J, Q, K, A, 2, と右に行けば行くほど強くなる.
- 初回はランダムに席順を決定し, カードの配布が行わ

[1] 電気通信大学大学院情報理工学研究科
Graduate School of Informatics and Engineering, The University of Electro-Communications
[2] 静岡県立大学経営情報学部
School of Management and Information, University of Shizuoka

れる．2 試合目以降は階級が与えられ, 大富豪にカードが配布された後, 席順にカードが配布されていく. ゲーム開始時のカードの枚数は 11 枚もしくは 10 枚となる．また, 席順は 3 試合ごとに, 階級に関係なくランダムに席替えが行われる.

- 最初に上がったプレイヤから順に, 大富豪, 富豪, 平民, 貧民, 大貧民と階級が与えられる. 次のゲームが始まる際, 与えられた階級に応じてカードの交換が行われる. 大富豪と大貧民は 2 枚, 富豪と貧民は 1 枚交換する．大富豪と富豪は相手に手札内の任意のカードを渡すことができる．大貧民と貧民は手札内の一番強いカードを渡す．カード交換の順序は貧民・大貧民によるカードの譲渡が先に行われ, その後富豪・大富豪が譲渡するカードを決定する．貧民と大貧民が譲渡するカードは, サーバー側でそれぞれの手札の中から強いカードを決定し, 自動的に交換が行われる．そのため, 貧民と大貧民はどのカードを譲渡したかはわからない.
- カード交換後, ダイヤの 3 を所持しているプレイヤからゲームが開始される．ただし, 必ずしも最初にダイヤの 3 を出す必要はない．プレイの順番は席順に沿って行われる.
- 自分に順番が回ってきたとき, 場に出ているカードより強いカードを提出しないといけないが, そのようなカードがないときは「パス」となり, 何も提出しない. また, 提出可能なカードがあっても, パスを選択することはできる．ただし, 一度パスを選択すると場が流れるまで自分の番にはならない.
- すべてのプレイヤがパスをすると, 場のカードがすべて除かれ, 最後にカードを提出したプレイヤに好きなカードを出す権利が与えられる．これを場が流れるといい, 場が流れた後最初に提出するプレイヤを親という．場が流れたとき, 最後にカードを提出したプレイヤが上がっていた場合は, 最後に提出したプレイヤの次の番のプレイヤが親となる.
- UECda で採用されているルールでは, 特定のカードで上がることを禁止する「禁止上り」というルールはなく, どのカードでも上がることができる.
- カードの提出方法 (役ということもある) は単体, ペア, 階段の 3 種類ある．単体はカードを 1 枚提出することである．ペアは同じ数字のカードを複数枚一度に提出することである．階段は, 同じスートで数字が 3 つ以上連続している場合に, 一度に提出することである．場にカードがない場合は, 3 つの提出方法のうち好きな方法で, 好きなカードを提出することができる．場に階段がある状況でカードを提出する場合は, その階段と同じ枚数で, なおかつ提出する階段を構成するカードの最小の数字が, 場にある階段を構成する最大の数字よりも強くないといけない.
- いずれかのプレイヤが 4 枚以上のペア, もしくは 5 枚以上の階段を場に提出した場合, 革命が発生し, カードの強弱は逆転する．この状態は場が流れても解除されず, 1 回のゲームが終了するまで継続される．また, 革命状態のときに再び革命が起こると, 革命状態は解除される.
- 場札のカードのスートの組み合わせと同じスートの組み合わせのカードが提出された場合, 縛りが発生し, 場が流れるまで同じスートのカードしか提出できなくなる．ペアの場合はすべてのカードのスートの組み合わせが同じでなければ, 縛りは発生しない.
- 8 のカードを提出すると 8 切りが発生し, 強制的に場が流れて 8 のカードを提出したプレイヤが親になる. これは, 8 のカードを含むペア・階段を提出した場合でも同様に発生する.
- 単体でジョーカーを提出するとき, ジョーカーは最強のカードとして扱われる．非革命時には 2 より強いカードとして, 革命時には 3 より強いカードとして使用することが可能である.
- ペアや階段を構成する際に, ジョーカーはあらゆるカードの代わりとして使用することができる．このとき, 8 の代わりとしてジョーカーを使ってペアや階段を提出した場合, 8 切りが発生する.
- スペードの 3 は通常時は単なる 3 のカードとして扱う. しかし, 場にジョーカーが単体で提出されている場合のみ, スペードの 3 を提出できる．このとき, 無条件で場が流れ, 提出者が親になる.

2.2 大貧民プログラム

UECda は実行時間によってライト級と無差別級の 2 つの階級に分かれている．ライト級はヒューリスティックな手法を用いたプログラムが主流で, 人間の用いる戦略をアルゴリズム化したものや, 複数の評価値の組み合わせで提出するカードを選択する手法が多い．一方, 無差別級では機械学習, 具体的にはモンテカルロ法を用いたプログラムが主流で, ランダムシミュレーションを用いた手法によって提出するカードを選択している．一般的に事前に学習されたパラメータを持つプログラムは, パラメータの値をランダムシミュレーション内での評価に使用しているが, ランダムシミュレーションと併用してパラメータをそのまま着手決定のための評価値計算に使用するものも存在する.

2.2.1 過去の大貧民プログラム

この項では過去の UECda に参加したプログラムのうちいくつか代表的なプログラムとその概略を説明する.

default

自分の手札と場のカードに関する簡単な条件分岐のみで着手を決定する UECda 標準プログラム．場が空の場合 (場にカードがない場合) は, 階段・ペア・単体の

順で優先順位が低くなり, 優先順位が一番高い役の中で一番弱いカードを提出する. 場にカードがある場合は, 場にあるカードより強いカードを提出するが, そのカードが手札内でペアや階段を構成するカードの一つである場合は, そのカードは提出しない. したがって, ペアや階段を崩さない.

kou2

田頭幸三氏によって作成された, 2015 年 UECda ライト級優勝プログラム [3]. 場にカードがないときの評価値である優先評価値, 手札全体の強さを表す手札評価値, 役の強さを表す強さ評価値の 3 つのヒューリスティックな評価値を組み合わせて着手を決定するプログラムである. ライト級プログラムでありながらモンテカルロ法を用いたプログラム並みの強さを持つ.

snowl

須藤侑弥氏によって作成された, 2010 年 UECda 優勝プログラム [5]. 2010 年は UECda では階級が分かれていなかったが, snowl はモンテカルロ法によるシミュレーションを用いているため無差別級相当の計算時間を必要とする. シミュレーションの効率を上げるためにシミュレーションの有望さとその評価の不確かさを組み合わせた値を定義し, その値が最大となる手を選択する UBC1-tuned というアルゴリズムを用いている.

Blauweregen

大渡勝己氏によって作成された, 2017 年 UECda 無差別級優勝プログラム. 事前学習パラメータを用いた評価値計算とモンテカルロ法を使用して着手を決定している. 場が空のときでのパスなど, 様々な状況に応じた戦略を考慮しており, 現状 UECda に出場しているプログラムの中では最も強いプログラムの一つである.

2.2.2 大貧民プログラム lilovyy

lilovyy は, 2017 年 UECda 無差別級の優勝プログラムである Blauweregen に対し, 開発者である大渡勝己氏が学習パラメータの再学習と戦略の改良を行ったプログラムである. まず相手がどのようなカードを持っていても絶対に勝てる手順である必勝手の探索を行い, 見つからなかった場合は方策関数とモンテカルロ法を用いて提出手を決定する.

lilovyy が着手決定の際に考慮しているルールやその組み合わせの詳細は以下のとおりである.

Joker やスペードの 3 を持つプレイヤ

Joker とスペードの 3 をそれぞれ自分か相手プレイヤのどちらが持っているかによって評価値が変わる.

手札のランクの平均

提出前と提出後の手札のランクの平均の大小で評価値が変わる. ここで, ランクとは強さのことである. 具体的にはカードの 3 がランク 1 で, 数字が大きくなるにつれてランクも増えていき, カードの K がランク 11, カードの A がランク 12, そしてカードの 2 がランク 13 である.

最小分割数

手札から構成できるカードの役の最小の組の数が提出前と提出後で変化するかで評価値が変わる.

Joker を使用した階段

提出後の手札に Joker が残る場合に行われ, Joker を用いた階段を構成できるかによって評価値が変わる.

着手の枚数

場が空の場合に行われ, 提出する着手の枚数によって評価値が変わる.

同じ枚数組の数

場が空の場合に行われ, 手札にある, 着手と同じ枚数の役の組の個数によって評価値が変わる.

縛り

場にカードがあり, かつ縛りをかけられる場合に行われ, 着手の枚数と縛り後のスートで一番強いカードを持っているかどうかで評価値が変わる.

革命

場が空, かつ着手で革命が起こせる場合に行われ, 自分の現在の階級によって評価値が変わる.

場が空のときのパス

場が空, かつ着手でパスを選択しているときに行われ, プレイヤ全員が場に提出していないカードの総枚数によって評価値が変わる.

パスでも次の親になることができる

自分以外のプレイヤがパスした後, かつ着手でパスを選択している場合に行われ, 他のプレイヤのカード枚数によって評価値が変わる.

パスしたときの親からの距離

着手でパスを選択している, かつ他プレイヤのカードの枚数が多い場合に行われ, 親からの距離, すなわち, 親から数えて何番目に提出するかによって評価値が変わる.

パスの後にプレイできる人数

着手がパス, かつ親がプレイ可能の場合に行われ, プレイ可能人数と親のカード枚数によって評価値が変わる.

スペードの３

場が Joker, かつ着手がスペードの 3 の場合に行われ, パスしたときに次に自分が親になるかどうかで評価値が変わる.

他にプレイできるプレイヤがいないときにカードを提出

他にプレイできるプレイヤがいない, かつ着手がパス以外の場合に行われ, 着手のランクによって評価値が変わる.

Joker 単体

着手が Joker 単体の場合に行われ, スペードの 3 を誰が持っているかによって評価値が変わる.

階段
着手が階段の場合に行われ, 場にカードがあるかどうかで評価値が変わる.

枚数組を崩す
場にカードがあり, かつ Joker を使用しないペア, もしくは階段を崩した単体またはペアが着手の場合に行われ, 着手のランクによって評価値が変わる.

空場での 8
場が空, かつ着手に 8 が含まれる場合に行われ, 残りの手札のカードのランクの組合せによって評価値が変わる.

8 切り
着手で 8 切りを起こすことができる場合に行われ, 提出後の自分の手札の枚数によって評価値が変わる.

最小ランクのカード
着手の選択時点で提出されていないカードの中で一番最小ランクのカードである場合に行われ, 革命かどうかで評価値が変わる.

着手と自身の残りの手札
着手と自身の残りの手札のランクとスートの関係性で評価値が変わる.

着手と相手の残りの手札
着手と相手全員の残りの手札のランクとスートの関係性で評価値が変わる.

3. 提案手法

事前学習されたパラメータを持つプログラムは, その挙動を理解するためには, コードのみだけではなく, パラメータまで読んで理解する必要がある. しかし, パラメータは数値の羅列であるため, その理解は非常に困難である.

本研究ではプログラムの挙動を理解するために必要な作業として, ルールやカードの枚数などの状況に対する評価値の大小の比較と, 着手とその他のカードの状況に対する評価値に関する規則を発見し, 理解することの 2 つであると考える. そのため, ルールやカードの枚数などの状況同士での評価値を比較する手法と, 着手と他のカードの状況の評価値の規則を見つけ, 数式に変換する手法を提案する.

3.1 重要視している状況

学習パラメータが大貧民におけるどの状況をより重要視しているかを明確にするために, 以下の手順で比較する.

(1) 着目している状況に関わる要素が, 別の状況と同じ要素を持っている場合には, それらの状況を統合する
(2) (1) で統合したそれぞれの状況において, 構成要素の状況の評価値の絶対値の平均をとる

ここで, 評価値の絶対値をとる理由は, 評価値がプラスマイナスにかかわらず, 値が大きいほどプログラムにとってそのルールの影響が大きいと思われるためである.

表 1 状況とそれに対する評価値

状況	評価値
パス	-0.9
空場でのパス	0.7
革命	0.25
8 切り	-0.1
Joker	0.3

例えば, 表 1 に示すような状況とそれに対応する評価値をもつパラメータがあるとする. このとき, 「パス」と「空場でのパス」を「様々なパス」という状況で統合する. また, 「Joker」と「8 切り」を「場を流す手」という状況で統合する. 次に, 統合した状況を構成する状況の評価値の絶対値の平均をそれぞれ計算すると, 表 2 のようになる. 表より, このパラメータは「パス」を重要視しており, 「場を流す手」はあまり重要視していないということになる.

表 2 統合した状況と評価値の絶対値の平均

状況	評価値の絶対値の平均
様々なパス	0.8
革命	0.25
場を流す手	0.2

3.2 パラメータの定式化

着手や残りのカードの枚数や, 誰が所持しているかなどの状況をよりわかりやすくするために, 以下の手順でパラメータから規則を見つけ, 数式に変換する.

(1) 事前学習されたパラメータを適用する状況によって分類する
(2) 状況ごとに分類した評価値を, 縦軸に値の大きさ, 横軸にカードのランクなどを用いたグラフ上にプロットする
(3) プロットした評価値の増減を近似直線で表すことによって数式化する

ここで, 近似直線で表す際に, 評価値の増減が著しく変化している場合には, その境界で区切ってその前後を別の数式で表して組み合わせる. 具体例として, パラメータをグラフ上にプロットすると図 1 のようになるとする. プロットしたグラフを見ると, 横軸の「5」でグラフの増減が著しく変化している. このとき, 横軸の「4」と「5」, または, 「5」と「6」の間を境界として区切って, 図 2 のような 2 つのグラフに分ける. これらのグラフに対して, それぞれ近似直線を求める. 図 2 から求められた 2 つの数式を, 図 1 のグラフを表す数式とする.

4. 重要視している状況に関する実験

4.1 実験概要と手法

提案手法を実際に大貧民プログラムに適用し, どのような分析結果が得られるか調べる. 本研究では提案手法を事前学習したパラメータを持つクライアントプログラムであ

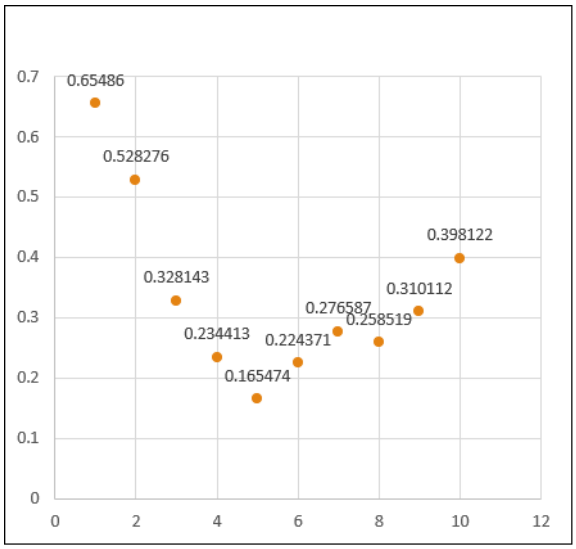

図 1　グラフ上にプロットされた評価値

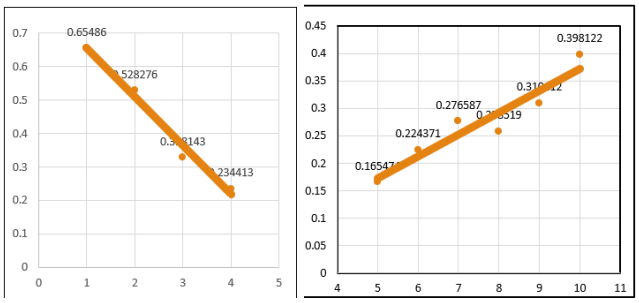

図 2　分割されたグラフと近似直線

る lilovyy に適用して実験を行い, 本実験では lilovyy の学習パラメータがどの状況を重要視しているのかを明確にする. ルールや状況によっては, 複数の状況を含む場合がある. その場合は複数の状況にそのルールの評価値を加算する. なお, 「着手と自身の残りの手札」と「着手と相手の残りの手札」は評価値の数と大きさが様々であり, 比較が困難であると判断したため, 対象から外した.

4.2　実験結果

以下に, 着目している要素を統合した状況ごとに列挙して示す.

パス

様々なパスに関する状況. 着手がパスであるならばここに含まれる.

- 場が空のときのパス：場が空のときのみ
- パスでも次の親になることができる
- パスしたときの親からの距離
- パス後にプレイできる人数

革命を起こす

革命を起こす手に関する状況. 場が空のときのみしか存在しない.

- 革命：場が空のときのみ

表 3　統合した状況と評価値の絶対値の平均 (場が空のとき)

状況	評価値の絶対値の平均
一度に多くの手札を減らす	6.045955
革命を起こす	3.61646
次に自分が親になる	3.45411
相手の提出を制限	1.024468
残りの手札を考慮	0.88255
パス	0.584257

表 4　統合した状況と評価値の絶対値の平均 (場が空でないとき)

状況	評価値の絶対値の平均
次に自分が親になる	2.792894
一度に多くの手札を減らす	1.753555
残りの手札を考慮	1.033868
パス	0.975724
相手の提出を制限	0.376729

次に自分が親になる

場を自分で流したりなど, 次に自分が親になる状況.

- パスでも次の親になることができる：場が空でないときのみ
- スペードの 3:場が空でないときのみ
- Joker 単体
- 8 切り
- 他にプレイできるプレイヤがいないときにカードを提出：場が空でないときのみ

一度に多くの手札を減らす

より多くの手札を減らすことを考慮している状況. 着手の枚数が多くなるほど絶対値が大きくなれば, ここに含まれる.

- 着手の枚数：場が空のときのみ
- 枚数組を崩す：場が空でないときのみ

残りの手札を考慮

着手提出後の手札を考慮した評価値計算を行う状況.

- Joker とスペードの 3 を持つプレイヤ
- 手札のランクの平均
- 手札の最小分割数
- Joker を使用した階段
- 空場での 8
- 8 切り
- 最小ランクのカード
- 同じ枚数組の数：場が空のときのみ
- 枚数組を崩す：場が空でないときのみ

相手の提出を制限

階段や縛りなど, 提出されるカードが制限される状況.

- 縛り：場が空でないときのみ
- 階段

統合された各要素の評価値の絶対値の平均値について, 場が空のときを表 3 に, 場が空でないときを表 4 に示す.

表 3 より, 場にカードがない場合は「一度に多くの手札

を減らす」が一番評価値が大きく,「パス」が一番評価値が小さかった. また, 表 4 より, 場にカードがある場合は「次に自分が親になる」が一番評価値が大きく,「相手の提出を制限」が一番評価値が小さかった. 以上の結果より, 場にカードがある場合と, 場にカードがない場合では, lilovyy が重要視している状況が違うことがわかった. また, 重要視している状況とあまり重要視していない状況が具体的にわかった.

5. パラメータの定式化に関する実験

5.1 実験概要と手法

事前学習されたパラメータに提案手法を適用して, 発見した規則を数式で表したとき, それを用いたアルゴリズムの強さが元の学習パラメータから大きく乖離していないかの確認を行うために, 計算機実験を行う. 本実験では提案手法を大貧民プログラム lilovyy に適用する.

まず, 各学習パラメータを, 以下に示すように, そのパラメータが表す, 着手と自分または相手のカードの枚数とスートの関係性ごとにまとめる. ここで,「着手と自分または相手のカードの枚数とスートの関係性」とは, 評価値を決定する着手が単体またはペアのときに着手の枚数およびスートと, 自分の残りの手札または相手の手札にある各ランクのカードの枚数およびスートとの関係性のことである. 例えば, 着手がハートとスペードの 4 の 2 枚ペア, かつ自分の残りの手札にハートの 6 の単体がある場合, 以下に示す「着手が 2 枚のとき, 着手に使用されるスートのカードが 1 枚」の状況にある着手と対象のカードのランクの評価値が適用される. さらに, 他の手札にあるカードを調べていき, 仮に 7 があるならば, 7 の枚数とスートを調べて, 着手のハートとスペードの 4 の 2 枚ペアとの関係を以下の関係性から適応する状況を見つけ, その評価値を適用する. また, この評価値は対象のカードが「自分の残りの手札」にあるか,「相手の手札」にあるかで変わる. 上記の具体例では「自分の残りの手札」にあるときの評価値が適用される.

(1) 着手が 1 枚のとき
- 着手と違うスートのカードが 1 枚のランク
- 着手と同じスートのカードが 1 枚のランク
- 着手と違うスートのみのカードが 2 枚のランク
- 着手と同じスートが含まれるカードが 2 枚のランク
- 着手と違うスートのカードが 3 枚のランク
- 着手と同じスートが含まれるカードが 3 枚のランク
- 4 枚のランク

(2) 着手が 2 枚のとき
- 着手に使用されないスートのカードが 1 枚のランク
- 着手に使用されるスートのカードが 1 枚のランク
- 着手に使用されないスートのカードが 2 枚のランク
- 着手と同じスートが 1 枚で, かつ違うスートが 1 枚のカードが 2 枚のランク
- 着手と全く同じスートのカードが 2 枚のランク
- 着手と同じスートが 1 枚で, かつ違うスートが 2 枚のカードが 3 枚のランク
- 着手と同じスートが 2 枚とも含まれるカードが 3 枚のランク
- 4 枚のランク

(3) 着手が 3 枚のとき
- 着手に使用されないスートのカードが 1 枚のランク
- 着手に使用されるスートのカードが 1 枚のランク
- 着手に使用されないスートを含むカードが 2 枚のランク
- 着手に使用されるスートのみのカードが 2 枚のランク
- 着手とは 1 枚スートが異なるカードが 3 枚のランク
- 着手と全く同じスートのカードが 3 枚のランク
- 4 枚のランク

(4) 着手が 4 枚のとき
- 1 枚のランク
- 2 枚のランク
- 3 枚のランク
- 4 枚のランク

lilovyy の評価値計算は, 合法手すべてに対して計算される. 例えば, 手札に 4 が 3 枚あるとする. このとき, 場が空の場合, 4 単体 3 通り, 4 の 2 枚ペア 3 通り, 4 の 3 枚ペア 1 通り, パスの合計 8 通りに対してそれぞれ評価値計算を行う. そのため,「着手と自分・相手の手札のカードの種類」の評価値については, 着手と同じランクのカードを持っていたときも評価値計算が行われる. しかし, 着手と同じスートで同じランクのカードが着手提出後の自分の残りの手札だったり, 相手のうち誰かが持っていたりすることはあり得ない. したがって,「着手と同じスートのカードが 1 枚」といった状況をはじめとする, 着手と同じスートのカードに関連したパラメータの中に「0」が並ぶことがある. ここで, lilovyy はまれな状況であったり, あり得ない状況であったりして, 学習されていないパラメータはすべて「0」で表される. したがって,「着手と同じランク未満のカード」と「着手と同じランク以上のカード」で分割することで, より規則性が見つかりやすくなると考えた. その後, 提案手法を分割したパラメータにそれぞれ適用し, 数式へと変換していく.

「着手と自分の手札のカードの種類」と「着手と相手の手札のカードの種類」に関する評価値には, それぞれ「非革命時かつ着手で縛ることができない」,「非革命時かつ着手で縛ることができる」,「革命時かつ着手で縛ることができない」,「革命時かつ着手で縛ることができる」の 4 つの場合に関する評価値が存在する. 本実験では「非革命時かつ着手で縛ることができない」と「非革命時かつ着手で縛ることができる」の 2 つの場合に関する評価値について提案手法を適用する. 本実験で使用する対戦プログラム

は以下のとおりである.

- poli: lilovyy の学習パラメータのみで対戦するプログラム
- poliX: poli の学習パラメータを提案手法で数式に変換したプログラム

実験の条件は以下に設定して行った.

- 試合数は 1 セット 10000 試合とし, 10 セット行う
- 試合のルールとして革命を起こさないことにする
- 対戦の組み合わせは poli 4 体と poliX 1 体で行う

5.2 実験結果

計算機実験の結果, 表 5 のようになった. 表 5 では, 4 体の対戦プログラム poli を便宜上, poli1 から poli4 と表記する.

表 5 計算機実験の結果

名前	poliX	poli1	poli2	poli3	poli4
1	30132	29980	29609	30130	30149
2	29721	30215	29859	29978	30227
3	29350	30040	30042	30214	30354
4	29801	30135	30463	30323	29278
5	29719	29989	29590	30355	30347
6	29696	30501	30208	29649	29946
7	29526	30149	29911	30687	29727
8	29988	30082	29474	30238	30218
9	29369	30572	29552	30251	30256
10	30209	29971	30071	29650	30099
平均得点	29751.1	30163.4	29877.9	30147.5	30060.1

表 5 より, poliX は poli よりも 10 セットの平均得点では若干弱くなっている. しかし, 1 セット目と 10 セット目では, 10000 試合を同じプログラム同士で対戦させた場合の理想的な平均得点である 30000 点を超えており, 10 セット目に至っては総得点が, 元のパラメータを使用した poli 4 つを抑えて 1 位になっている. 最少得点も, poli4 の 4 セット目が最小であり, 全体的に大きく負けているというわけではない. 大貧民は不完全情報ゲームという性質上, 運という要素がゲームの結果にかかわってくることも多いが, 10000 試合という試合数においては, 他のプログラムとほぼ同等の強さを持たなければ 30000 点を超えることはない. したがって, 本実験結果は, 学習パラメータを提案手法に従って数式に変換した戦略によって挙動を決定しても, 強さはあまり変わらないことを示唆している.

6. 考察

6.1 重要視している状況に関する実験

表 3 および表 4 より, 一度に多くの手札を減らす状況は, 場にカードがないときは圧倒的に重要視され, 場にカードがあるときでもかなり重要視されている. これは, 大貧民の手札を早く 0 枚にしたほうが勝つという性質上, 重要視されるのは当然であると考えられる. 弱いカードは場にカードがある場合にはほとんど提出できず, 革命が起きて強いカードになることを待つか, 親になって提出するかのどちらかになるのがほとんどである. そのため, 場が空のときに他の状況に比べて特に大きくなっているのは, 場が空のため, 弱いカードを一度に多く提出できるという点が大きいと考えられる. つまり, 場が空のときに枚数を増やすことで, 相手にあまり提出させない, もしくは相手の強いカードを一度に多く提出させることができるため, 重要視されていると考えられる. ペアは 2 枚, 3 枚, 4 枚と枚数が増えるにつれて手札に存在する確率が小さくなる. そのため, 枚数が多いほど次に親になることができる確率も上がり, なおかつ自分の有利な状況を続けることができる.

次に自分が親になる状況も, 場が空のとき, 場にカードがあるとき両方の場合で重要視されている. 特に, 場にカードがあるときには 2 番目に重要視されている状況の評価値が 1.7,「次に自分が親になる」が 2.7, と他の状況と比べて重要視している. これは, 大貧民において親になることで自分に有利な状況を自分で作り上げることを重要視しているためだと考えられる.

相手の提出を制限する状況は, 場が空のときは平均的に重要視されているが, 場にカードがある場合はほとんど重要視されていない. 場が空のときと場にカードがあるときの違いは, 縛りである. つまり, 縛りの評価値を入れることで, ここまでの差が出るということである. 縛りは, 場合によっては有効かもしれないが, 場を縛った次の自分の手番で提出できないなどのデメリットもあり, かつ相手に与える制限がスートのみで, あまり大きな制限を与えられていないため, 評価値が小さくなっていると思われる. 場が空のときは, 階段のみであるが, 階段は自分の手札の弱いカードも提出できる, 相手の連続したランクのカードを消費させられるといったメリットが大きいと判断されたため評価値が大きくなっていると考えられる.

一方, パスは場が空のとき, 場にカードがあるとき両方の場合で重要度が低い. パスはその手番だけでなく, 場が流れるまで提出できなくなる. そのため, 他プレイヤとカードの枚数差が大きくなってしまうので重要度が低くなっていると考えられる.

6.2 パラメータの定式化

表 5 より, 変換した数式を用いたプログラムは元のプログラムより若干弱くなったが, 本実験では変換が妥当と判断した. 若干弱くなった原因は, 近似直線を用いたことによる元のパラメータとのずれが原因と考えられる. lilovyy におけるパラメータの評価値は 0.001 違うと結果が変わってくる.

また, 元のパラメータからずれている部分の状況と, その

パラメータが表す状況がゲーム中に現れる回数も強さの変化に関わってくると考えられる．具体例を挙げると, 元のパラメータから 0.01 ずれ, その状況が 1 ゲーム中に 100 回現れた場合, 元のパラメータを用いたプログラムから 1 ずれることになる．また, 元のパラメータから 0.1 ずれていたとしても, その状況が 1 ゲーム中に 1 回しか現れない場合, 元のパラメータを用いたプログラムから 0.1 しかずれない．実際に, lilovyy のパラメータ評価値における, 「着手と自分の手札のカードの種類」での「非革命時」かつ「着手で場を縛ることができる」ときに, 「着手 4 枚で自分の手札に対象のカードが 4 枚ある」場合の評価値は, その状況が希少であるため, 学習がされていない部分が多い．具体的には, 「着手のランク 13 種類」×「対象のカードのランク 13 種類」 の最大 169 箇所のパラメータに対し, 実際に学習されているのは 17 箇所で, 約 10% しか学習されていない．これは, 「着手 4 枚で自分の手札に他に何か 4 枚ペアがある」という確率の低い状況に合わせて, 「着手で縛ることができる」, つまり, 「場に 4 枚ペアが提出されている」という状況も起きていることになる．また, 本研究では革命が起きない設定で実験を行ったが, lilovyy の学習時には革命が起きる設定で学習を行っている．つまり, 学習時には上記の状況に加えて, 「非革命時」つまり「革命が起きている状況で 4 枚のカードが提出され, 革命状態が解除された」という状況が起きている．このような状況が組み合わさることは非常にまれであるため, 学習されていない部分が発生していると考えられる．以上のことから, 本研究では行っていないが, 1 ゲーム中に各状況が何回現れたかを計測し, 回数が多い場所を中心により細かな数式変換を行うことで, 元のパラメータに強さを近づけることができると考えられる．

7. おわりに

7.1 まとめ

本研究では, 事前に学習されたパラメータから一定の規則を見つけ, 数式に変換するために, パラメータによって考慮されている状況をまとめ, 評価値を比較する手法を提案し, 実際の大貧民プログラム lilovyy に適用して検証を行った．その結果, lilovyy は「一度に多くの手札を減らす」ことを重要視し, 「パス」と「相手の提出を制限する」ことはあまり重要視していないことが分かった．また, 評価値を適用する状況を統合して, グラフ上にプロットし近似直線を用いて定式化する手法を提案し, 実際の大貧民プログラム lilovyy で検証を行った．その結果, 変換した数式によるプログラムは元の学習パラメータを用いたプログラムと強さにおいて大きな変化はなく, パラメータの規則を見つけ, 数式として表現できるという結果が得られた．

7.2 今後の課題と展望

本研究では, 定式化に近似直線を用いたが, そのことによるずれが多少発生した．これの解消のために, 各状況が 1 ゲーム中に何回発生するかの頻度調査を行い, その頻度が多いところを中心にずれを解消する数式への変更を行うことで, より詳細な定式化ができると思われる．また, 本研究を応用して, 自然言語で説明できる戦略を数式やグラフから発見する手法を確立することで, パラメータから自然言語で説明可能な戦略を導くことができたり, より簡単な数式を用いてパラメータを変換することができるようになると思われる．

参考文献

[1] David Silver, Demis Hassabis, AlphaGo: Mastering the ancient game of Go with Machine Learning, https://ai.googleblog.com/2016/01/alphago-mastering-ancient-game-of-go.html 参照 2021-1-19
[2] 電気通信大学, UEC コンピュータ大貧民大会, http://www.tnlab.inf.uec.ac.jp/daihinmin/2020/ 参照 2021-1-19
[3] 田頭 幸三, 但馬 康宏, コンピュータ大貧民におけるヒューリスティック戦略の実装と効果, 情報処理学会論文誌, Vol. 57, No. 11, pp. 2403-2413 (2016).
[4] UEC 標準ルール, http://www.tnlab.inf.uec.ac.jp/daihinmin/2020/document_rules.html 参照 2021-1-19
[5] 須藤 侑弥, 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「snowl」の開発, 第 2 回 UEC コンピュータ大貧民シンポジウム資料, 2010.
[6] 人狼知能プロジェクト, http://aiwolf.org 参照 2021-1-19
[7] Junjie Li, Sotetsu Koyamada, Qiwei Te, Guoqing Liu, Chao Wang, Ruihan Yang, Li Zhao, Tao Qin, Tie-Yan Liu, Hsiao-Wuen Hon, Suphx: Mastering Mahjong with Deep Reinforcement Learning, https://arxiv.org/abs/2003.13590