**■ ポーカーの戦略プログラミング**

**● トランプの内部表現**

トランプ一式52枚のカードを表すのは、0から51までの整数を用い、整除演算での対応を考える。

この整数の値をトランプのカードの識別番号(コード)という。

図表のように、13で割った整商がマークの種類、剰余が数位となるように対応させればよい。

ここで、スペードS‥0、ハートH‥1、ダイヤD‥2、クラブC‥3 とする。

また、剰余kを数k+1に対応させる。ただし、A‥0、T‥9、J‥10、Q‥11、K‥12 とする。

すなわち、種類コードm、数位コードnに対し、識別番号 t＝m×13＋n を与えることができる。

識別番号tから種類コードmと数位コードnを求めるには、整除演算 t÷13＝m‥n を用いる。

すなわち、t/13＝m, t%13＝n となる。例えば、ハートのクイーン(HQ)は、1×13＋11＝24 より、24で表される。

整数47は、47÷13＝3‥8 より、クラブの9を表す。また、2枚のカードについて、13で割った整商が等しければ

同じ種類であり、4で割った剰余が等しければ同じ数位であることが分かる。

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **A** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** | **J** | **Q** | **K** |  |  |
| **スペード** | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | **S** | 刀剣を表し、軍人を象徴 |
| **ハート** | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | **H** | 聖杯を表し、僧侶を象徴 |
| **ダイヤ** | 26 | 27 | 28 | 29 | 30 | 31 | 32 | 33 | 34 | 35 | 36 | 37 | 38 | **D** | 宝石を表し、商人を象徴 |
| **クラブ** | 39 | 40 | 41 | 42 | 43 | 44 | 45 | 46 | 47 | 48 | 49 | 50 | 51 | **C** | 棍棒を表し、農民を象徴 |
|  | **A** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **T** | **J** | **Q** | **K** |  |  |

**● ポーカーのルール**

トランプのポーカーゲームは、5枚の手札で以下のような9種類の役を作るゲームである。高い配点の役を作った方が勝ちである。

役無しは、0点である。同時に複数の役ができているときは、高い役を配点とする。役の内容は、影響しない。

同位札は、数字が同じ。

同種札は、種類が同じ。

順位札は、数字が連続。

Aは2にもKにも繋がる。

ただし、K,A,2 とは繋がらない。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 番号 | 役名 | 札構成 | 例 | 得点 |
| 0 | ノーペア | 役無し | SA H3 D2 C4 CQ | 0 |
| 1 | ワンペア | 2枚の同位札1組 | H5 D5 S2 S6 CK | 1 |
| 2 | ツーペア | 2枚の同位札2組 | S5 D5 H4 S4 C7 | 2 |
| 3 | スリーカインズ | 3枚の同位札1組 | SK DK CK HJ H2 | 8 |
| 4 | ストレート | 5枚の順位札1組 | SA D2 H3 C4 D5 | 32 |
| 5 | フラッシュ | 5枚の同種札1組 | C2 C5 C7 CT CK | 24 |
| 6 | フルハウス | 3枚の同位札1組と2枚の同位札1組 | SA S3 SQ D3 D8 | 16 |
| 7 | フォーカード | 4枚の同位札1組 | S7 H7 C7 D7 S2 | 64 |
| 8 | ストレートフラッシュ | 5枚の同種順位札1組 | S8 S9 ST SJ SQ | 128 |
| 9 | ロイヤルストレートフラッシュ | 10から始まるストレートフラッシュ | ST SJ SQ SK SA | 256 |

ポーカーのルールには、手札を取り替えないスタッド式と、取り替えるドロー式とがある。手札5枚のドロー式で、1枚だけの手札の交換が

何回か可能なゲームを考える。1枚も取り替えないことも可能とする。カードは、ジョーカー無の52枚を使う。本演習では、確率を考慮

したヒューリスティックな戦略を目的とするため、同じ山札で何回かポーカーを続けるゲームを扱う。既に使用した山札(場札や捨札)を

記憶しておき、残りのカードの傾向を推定して、より可能性の高い手を狙うようにする。ここで、以下のように、用語を整理する。

本演習では、スワップ数は1に固定する。チェンジ数とテイク数を指定することにより、ゲームのパラメタが決まる。

例えば、ポーカー 02\_3 は、チェンジ数 2、テイク数 3の場合である。

○ スワップ数 一度に交換する枚数

○ チェンジ数 交換を行なう回数

○ テイク数 同じ山札でポーカーを続ける回数

○ トライ数 以上のようなゲームを繰り返す回数

ただし、山札の枚数52による制限がある。ゲームの途中で山札がなくなったら、そこでチェンジは打切である。

また、手札5枚が確保できなければ、ゲームとして成立せず、そのテイクは0点となる。

例えば、ポーカー 05\_6 では、5枚の手札で5回のチェンジを行うと、1回のテイクで10枚を使う。すなわち、テイク5までで

最大50枚を使ってしまう。このとき、残りは2枚となるので、テイク6はゲームとして成立せず、0点となる。

したがって、テイク5までで最大47枚しか使えない。また、最後のテイクでは、残り枚数までしかチェンジできない。

さらに、最近は、テイクごとの重みを付け、傾斜得点とする。例えば、テイク3までの重みが、1.0, 1.5, 2.0 であれば、

テイク2は素点の1.5倍、テイク3は素点の2.0倍の得点として、加算される。

**● 2017年度のルール**

2017年度の大会規定は、ポーカー 05\_6 で、傾斜掛率は 1.0, 1.5, 2.0, 1.0, 1.5, 2.0 とする。

(5＋5)×6＝60 なので、チェンジ数が多いと、山札が足りなくなる。

**● 戦略のヒント**

ポーカーの手役は、ペア系、ストレート系、フラッシュ系に大別される。ストレート系とフラッシュ系は、5枚の手札が

全て絡むので、配点は高いが、0点となってしまう危険性がある。ペア系は、既にできたペアを残しながら、他のカードを

交換していけるので、安全確実であるが、配点は低い。したがって、いつ高配点のストレート系やフラッシュ系を狙うか、

どのカードを切って待つか、どこで諦めてペア系に切り替えるかが、戦略の基本方針となる。場合によっては、既にできて

いるペアを崩して高得点を狙うことも考えられる。また、得点の割にできやすいフルハウスがポイントになる。

ストレートについては、3,4,5,6,X の状態で 2,7 を待つ両面形、3,4,5,X,7 の形で 6 を待つ嵌張形、A,2,3,4,X の形で 5 を

待つ辺張形がある。さらに、テイク数が進めば、もうハートは残り少ない、3の札はまだ1枚も出ていないなどの情報を使って、

より緻密な戦略を練ることができる。なお、フォーカード以上は、非常に確率が低く、狙ってできるものではない。期待値と

しても、割に合わないので、できれば儲けものと捉えた方がよい。

傾斜配点に対しては、重みの高い最後の2回のテイクには、十分な枚数を残しておく必要がある。逆に、重みの低い最初の

2回のテイクは捨てゲームと考えてもよい。少ないチェンジで高い役が期待できないようであれば、ノーペアでもチェンジを

打切にした方がよい。テイクごとに、チェンジ数の目安を想定し、序盤、中盤、終盤と、それに最適な戦略を切り替える

ようにする。

**● 例題プログラムのフォルダ構成**

例題プログラムのフォルダ構成は、以下のようになっている。演習中の戦略名は適当でよいが、最終成果として

提出するときは、s15t200\_01, s15t200\_02 のように、アカウント名と番号2桁の形式に揃える。

インスタンス名も、s15t200\_01\_03 のように、戦略名\_番号(2桁)の形式とする。

usr/

s15t200\_00/ ユーザ入力

s15t200\_01/ 交換せずにすぐ打切とする

s15t200\_02/ 常に左端のカードを捨てる

s15t200\_03/ スリーカインド以上ができるまで交換

s15t200\_04/ テイクによって戦術を変更

sys/ ツール

DeckInit 山札生成プログラム

BinCompile.sh コンパイルスクリプト

PlayUser.sh ユーザ実行

PlayComp.sh コンピュータ実行

src/ ソース

Poker.h ヘッダ

PokerExec.c ポーカー実行

CardLib.c カード処理

DeckInit.c 山札生成

DeckConvert.c 山札変換

usr/ ユーザプログラム

s15t200\_00/ 戦略フォルダ(例示)

PokerOpe.c 戦略

user\_init/ 戦略フォルダ(ユーザ対話実行)

戦略名/ 戦略フォルダ

log/ ログ

s15t200\_00\_00/ インスタンス(例示)

Deck.ini 使用山札

Result.txt 実行結果

Game.log 途中経過

インスタンス名/ インスタンス

doc/ ドキュメント

tmp/ 一時フォルダ

**● 戦略の実装**

演習では、戦略に従って、手札 hd[]、場札 fd[]、チェンジ数 cg、テイク数 tk、捨札 ud[] 捨札数 us、を入力とし、

交換したい出札を選ぶ関数 strategy() を作成する。返却値は、出札とする手札の配列添字(0～4)である。

手札を交換しないときは、-1 とする。この場合、以降の交換を打ち切ったことになる。

例題プログラムは、PokerOpe.c のみを編集すればよいようになっている。

位数または種類で理札する、聴役状態(後1枚で役が完成)の判定など、必要な補助関数も定義する。

どの手札でもよい、どちらの戦略でもよいというときは、0以上1未満の実数乱数を用いて、確率的に混合戦略を決める。

int strategy(int hd[], int fd[], int cg, int tk, int ud[], int us) {

//---- 本体処理

if ( poker\_point(hd) >= P3 ) { return -1; } // P3以上の役ができていれば、手役を確定させる

if ( tk < 2 && cg >= 3 ) { return -1; } // テイク1と2では、3回までしか交換しない

//---- 返却処理

return 0;

}

**● 実行結果の確認**

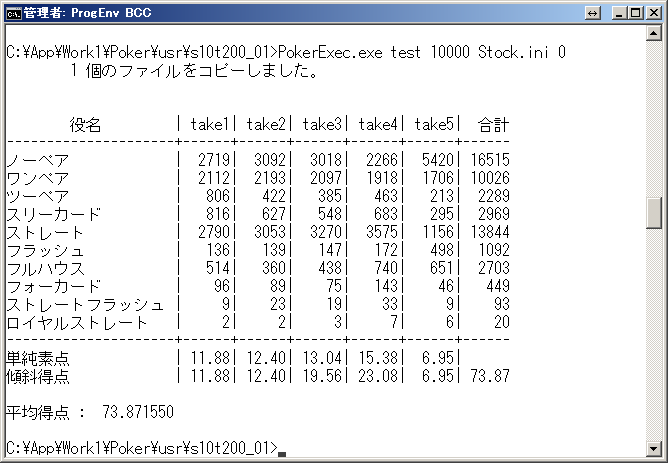
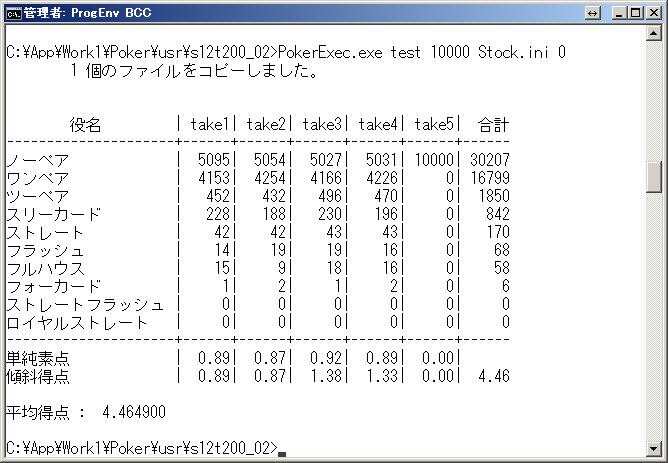
異なる乱順列から生成された山札で、10000回ゲームを繰り返し、平均得点を求める。各役の頻度分布も求めて、戦略を評価せよ。なお、手札を配布されたままの状態でも、約40%の確率でワンペアが

できている。また、約10%の確率でツーペアもできる。したがって、1回も交換しないという戦略でも、

1テイクの平均素点は0.9点程度になる。傾斜掛率も考慮した5回のテイクの合計は、約6.3点である。

以下の左の例では、カードを使いすぎ、5テイク目が打切りで0点になっているため、得点が低くなっている。

戦略の方向性としては、高い配点(ストレートやフラッシュ)の役ができる代わりに

ノーペアも多いリスク許容型と、ノーペアも高い配点も少ないリスク回避型がある。

2012年度の最強プログラムを、2016年度の規定で実行したところ、右の例のように70点程度となった。

ノーペアが半減し、ストレートが大幅に増えている。

これは、傾斜配点と山札不足に対応していない。それも考慮したとすると、30～40%は向上すると予想される。

したがって、100点が目指すべき目標である。

**● 実行環境**

2015年度から、実行環境を UNIX 上に移行し、言語処理系として GCC を標準とした。実験１で使った chausson0 や gin あるいは、

各研究室のLinuxサーバにログインして、開発を進める。もちろん、自前のLinuxを用いてもよい。

言語処理系 GCCは、標準で導入されているはずである。

Mac 上では、Clang を使う。MS Windows で開発を行うには、CygWin その他の環境ツールが必要である。仮想環境を導入してもよい。

**● 大会運営サーバWinT**

大会運営サーバ WinT は、富永研究室の教育支援サーバ chausson3.eng.kagawa-u.ac.jp 上で稼働している。

授業Webからもリンクを貼っておくが、URLは、以下の通りである。アクセスするWebブラウザは、 IE 以外を推奨する。

なお、サーバの障害時には、chausson7 を代替とすることがある。

http://chausson3.eng.kagawa-u.ac.jp/App/GameExrc/WinT/login

提出するのは、ソースコードである。バイナリではない。必ず、ローカルに実行を確認してから提出する。

**● 演習の作業手順**

usr フォルダ下で、戦略ごとにフォルダを作成し、PokerOpe.c 中の関数 Strategy() をコーディングする。

sys フォルダ下のバッチファイルを使い、ゲームのパラメタ(チェンジ数、テイク数)を指定して、コンパイルする。

実行においては、まず山札ファイルを生成する。次に、インスタンス名、トライ数を指定して、実行する。

このとき、log フォルダに、インスタンス名のサブフォルダが作成され、その下に実行結果が格納される。

大体、以下のような作業手順となる。

%% 初期作業

共有フォルダから例題プログラムをダウンロード

例題プログラムのルートフォルダに移動

%% コーディング

**cd usr**

**mkdir s15t200\_01**

**cd s15t200\_01**

**cp ../s15t200\_00/PokerOpe.c PokerOpe.c**

**emacs PokerOpe.c**

テキストエディタを起動して戦略ソースのコーディング

%% コンパイル

戦略ソース(PokerOpe.c)をパラメタを指定してコンパイルする

**../../sys/BinCompile.sh ライブラリ指定 チェンジ数 テイク数 戦略ソース**

**(実行例) ../../sys/BinCompile.sh ../../src 6 6 PokerExec.c**

2017年度の演習用にパラメータを設定した下記のスクリプトを使ってもよい

**../../sys/Make2017.sh**

各組の課題用のシェルスクリプトを用意している

%% 実行と確認

**../../sys/DeckInit 山札数 > 山札ファイル**

**(実行例) ../../sys/DeckInit 1000 > Deck\_01.txt**

実行結果の保存先(インスタンス)を指定して実行する

**PokerExec インスタンス名 トライ数 山札ファイル 表示モード(0/1)**

**(実行例) PokerExec s15t200\_01 1000 Deck\_01.txt 0**

インスタンス名を実行ごとに変更しないと上書きされてしまう

またはゲーム型を指定済のバッチコマンドを使ってもよい

**../../sys/PlayComp.sh**

**cd ../../log/インスタンス名**

テキストエディタで得点結果 Result.txt や実行状況 Game.txt を開いて確認する

新しい戦略を実装したら、少ないトライ数の山札ファイルに対して実行し、意図通りの動きをしているかをまず確かめる。

正しく動作しているようだったら、ある程度の量の山札ファイルに対して実行し、パラメタを調整する。

なお、同じ山札ファイルで、何度も実験することもできる。このとき、インスタンス名(s15t200\_00\_01, s15t200\_00\_02)を

換えて、別フォルダにログを残すようにする。

● 作業

まず、usr/ 下のフォルダ s15t200\_00 ～ s15t200\_04 を、コンパイルして実行してみる。

次に、各フォルダ を自分のアカウント名でコピーする。何らかの変更を行う。

プログラムには、アカウントと氏名を明記する。プログラムの愛称や特徴、作成日時も書いておく。

大会運営サーバ WinT に提出して、プログラムが受理され、得点が表示されるか確認する。

**■ 課題レポート第１回**

オンラインの提出先は、**\\chausson0\Report\2017\InfoExpr2\Report1\** のアカウント名の下で、主ファイル名は **Report1** とする。

答案のテンプレートとして、**\\chausson0\Material\2017\InfoExpr2\Problem\Style1.docx** を利用する。

戦略コードのプログラムとしての配点は、60点である。最終大会の戦績により、順位と得点(達成度)において、得点が決まる。

課題レポートの配点は、90点である。

**● 課題1-1 問題分析と基本戦略 [20点]**

**〔問題分析〕 5点**

ポーカーの役の確率など、純粋なゲームとしての性質を数学的に分析する。結果の数値だけでなく、計算過程も明記する。

ランダムに選んだ5枚からできる手役の確率は、既に書籍やWebページに幾らでも掲載されている。それを引用してもよい。

さらに、以下のような典型的な場面について、確率を考察せよ。

・ 4枚の種類がハートであるとき、残りの山札がn枚で、うちハートがm枚とする。c回の交換までに、フラッシュができる確率

・ Aと2の数位でツーペアができているとき、残りの山札がn枚で、うちAと2がm1,m2枚とする。c回の交換までに、フルハウスができる確率

・ 2,3,4,5の4枚を持っているとき、残りの山札がn枚で、うちAと6がm1,m2枚とする。c回の交換までに、ストレートができる確率

・ A,2,3,4の4枚を持っているとき、残りの山札がn枚で、うち5がm枚とする。c回の交換までに、ストレートができる確率

・ A,2,4,5の4枚を持っているとき、残りの山札がn枚で、うち3がm枚とする。c回の交換までに、ストレートができる確率

**〔基本戦略〕 10点**

また、経験的な傾向を掴む。実際にトランプカードで遊んでみるとよい。WinTのローカル実行環境も使ってみる。

ヒューリスティックな知識から、どの手札を交換するかの戦略を考える。個々のルールは、以下のようなプロダクションルールで考える。

・ ストレート以上なら、手札を交換しない。

・ スリーカインズならフルハウスまたはフォーカードを狙う。

・ ツーペアならフルハウスまたはスリーカインズを狙う。

・ ワンペアならスリーカインズまたはツーペアを狙う。

・ ノーペアで、ストレートまたはフラッシュの聴札なら、それを狙う。

・ ノーペアで、それ以外のとき、ワンペアを狙う。

次に、これらのルールをどう組み合わせるか、優先度をどう付けるか、図解を交えて説明する。フローチャートなどを用いるとよい。

手役ができていたら打切とし、手役ができるまで手札を交換するというには、当たり前のことなので、基本戦略を述べたことにならない。

手役の一歩手前や二歩手前で、どういう手役を狙って、どの手札を捨てるかを記述する。

交換枚数によって、判断が変わってくる。残りの山札の状況や、テイクの掛率も考慮しなければならない。

**〔作業計画〕 5点**

作業計画を立てる。実現目標を掲げる。

開発環境(OSや言語処理系)を述べる。

バージョン管理やテスト手法のツールも紹介する。

**● 課題1-2 プログラムの解説 [30点]**

**〔全体設計〕 10点**

最終大会に提出し、最高得点を得た最良戦略について、詳細に説明する。

各自が定義した関数の原型宣言の一覧を挙げ、仮引数と返却値の意味や副作用を説明する。

各関数の定義本体で、どの関数を参照しているかのクロスリファレンスを挙げる。

大域変数や型定義などもあれば挙げておく。

プログラムの締切後に、より強い戦略を作成した場合は、それについても述べてよい。

**〔詳細実装〕 10点**

各関数のうち、ソースコードの重要な部分を行番号を付けて抜き出し、実装上のポイントを説明する。

特に、概念的な基本戦略がプログラミング的にどう実装されるのかが分かるようにする。

効率の良い簡潔なコードになっているか。

なお、予備大会の戦略コードでは実装したが、最強戦略では使わなかったメソッドについても、アピールすべきものは述べてよい。

**〔動作検証〕 10点**

重要な機能について、関数単位での単体テストを、どう実施したか。

基本戦略で考えたプロダクションルールにおいて、特定の山札による実行が適切に動作しているか。

先読が制限時間内で正しく動作しているか。

**● 課題1-3 予備大会と実験状況 [15点]**

**〔予備大会〕 10点**

予備大会での提出コードの得点状況について概観する。5つ程度の主要な戦略について説明する。

いつの時期、どういうプロダクションルールの戦略で、どのバージョンのコードが、どういう成績であったか。

ローカルに生成した山札のサンプルに対する戦譜(手役と捨札の状況)の重要部分も挙げる。

**〔実験状況〕 5点**

予備大会の期間中、戦略コードのバージョン管理をどのように行ったか。

コーディング、デバッギング、テスト、アップロードなどで、便利なツールをどう使ったか。

あるいは、プログラムの自動実行、サンプルデータの管理などで、自分で便利なスクリプトを書いて工夫したか。

**● 課題1-4 最良戦略と結果考察 [25点]**

**〔最良戦略〕 15点**

最良の結果を得た実行画面を画像ファイルとして1つ載せる。それ以外の実行結果については、図表で整理して示す。

締切後に、より強い戦略を作成した場合は、それについても述べてよい。

ペア系、ストレート系、フラッシュ系の割合、テイクごとの得点の傾向などを調べる。

**〔結果考察〕 10点**

予備大会および最終大会の状況も踏まえ、自分が作成した提出コードについて、総合的に考察する。

検討したアイデア、プロダクションルール、設定パラメタ、先読手法は、適切であったか。

実装は、要件や仕様の通りに行えたか。バグは発見できたか。

効率的な実装になっているか。リファクタリングが行えたか。

戦績や戦譜には、どんな特徴や傾向があったか。反省すべき点や残された点は何か。

問題解決へのアプローチとして、課題への取組の状況は適切であったか。

**〔更新状況〕 10点**

予備大会での戦略プログラムの更新過程について述べる。どのようにして、適切なコード品質を目指したかを述べる。

主なバージョンごとに、コードの概略を説明する。

新しい戦術の導入、戦術の方針転換、パラメタの調整、バグの修正、コードの整理、関数への分割など。

WinTサーバで示される、ABCサイズなどのコード指標の推移についても述べる。