Este algoritmo tiene como objetivo calcular las probabilidades de tener una mano más fuerte que la del oponente, una mano similar a la del oponente o una mano peor que la del oponente; y con esa probabilidad, tomar una de las tres posibles acciones: ver la apuesta, de subir la apuesta o de pasar la apuesta. Para ello, tenemos que tener en consideración siempre las cartas de las que tenemos información:

**Información común para todas las fases de la ronda**

Durante cada fase de la ronda, siempre vamos a tener los siguientes datos:

* Nuestra mano: dos cartas, cada una con un número y un palo, que son dos cartas conocidas.
* La mano del oponente: dos cartas, al igual que nuestra mano, pero en este caso son cartas que van a ser desconocidas siempre hasta el final de la ronda.
* Cada carta es única, es decir, una carta no puede aparecer dos veces

Con estos 3 datos, podemos ver que tenemos una incógnita que debemos suponer de cara a calcular las probabilidades arriba mencionadas, pero el tercer dato (la unicidad de las cartas) nos permite suponer cada una de las posibles manos del oponente y, con eso, poder calcular las probabilidades.

**Tablas de probabilidades: tabla de pesos relativos y tabla de probabilidad de acción del oponente**

Para todo el funcionamiento del algoritmo vamos a partir de dos tablas: una tabla que va a recoger la probabilidad relativa de que el oponente tenga una mano y una segunda tabla que recoja la probabilidad de cada mano del oponente de tomar una decisión u otra.

La tabla de pesos relativos recoge todas las posibles combinaciones de cartas y un valor numérico, al que denominaremos Peso relativo (Pr), que puede tener valores entre 0 y 1. Este valor representa cómo de probable es que el oponente tenga una mano y otra y se irá actualizando con la decisión que tome el oponente.

La tabla de pesos relativos inicialmente tendrá estos valores.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | TT | TP | TD | TC | PT | PP | PD | PC | DT | DP | DD | DC | CT | CP | CD | CC |
| AA | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| KK | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| QQ | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| JJ | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| 1010 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| 909 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| 808 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| 707 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| 606 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| 505 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| 404 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| 303 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| 202 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| AK | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| AQ | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| AJ | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| A10 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| A09 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| A08 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| A07 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| A06 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| A05 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| A04 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| A03 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| A02 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| KQ | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| KJ | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| K10 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| K09 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| K08 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| K07 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| K06 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| K05 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| K04 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| K03 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| K02 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| QJ | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| Q10 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| Q09 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| Q08 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| Q07 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| Q06 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| Q05 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| Q04 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| Q03 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| Q02 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| J10 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| J09 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| J08 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| J07 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| J06 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| J05 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| J04 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| J03 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| J02 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 1009 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 1008 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 1007 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 1006 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 1005 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 1004 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 1003 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 1002 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 908 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 907 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 906 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 905 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 904 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 903 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 902 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 807 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 806 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 805 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 804 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 803 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 802 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 706 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 705 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 704 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 703 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 702 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 605 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 604 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 603 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 602 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 504 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 503 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 502 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 403 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 402 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 302 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

Representando cada fila una combinación de dos cartas y cada columna el palo (T= tréboles, P= picas, D=Diamantes y C=Corazones) de cada una de las cartas. Por ejemplo, si cojemos el valor de la fila 807 y la columna CD, ese valor corresponde a la probabilidad relativa de que el oponente tenga un 8 de corazones y un 7 de diamantes.

Como podemos observar, hay varias combinaciones que siempre tendrán una probabilidad de 0. Estas combinaciones ocurren únicamente en las parejas ya que:

* Al ser las cartas únicas, no puede haber dos cartas iguales. Por tanto, una pareja de dos cartas del mismo palo (por ejemplo, tener dos 9 de diamantes) es imposible. Por tanto, estos casos siempre tendrán probabilidad de 0.
* Al ser parejas, el orden de las cartas no influye. Es decir, tener un As de diamantes y un As de picas es lo mismo que tener un As de picas y un As de diamantes. Por esta razón, a las posibilidades repetidas, se les ha dado valor 0 para evitar valores duplicados.

Para actualizar estos valores, se tiene la segunda tabla, la tabla de probabilidad de acción del oponente.

También cada vez que una carta sea conocida, todas las combinaciones posibles con esa carta se convertirán en 0.

Antes de hablar de la tabla, vamos a introducir un concepto que tendrá bastante relevancia durante este algoritmo y es la probabilidad de acción triple. Esta probabilidad de acción es un conjunto de tres valores entre 0 y 1 {p, v, s}, que representan la probabilidad que tiene de pasar la apuesta, de ver la apuesta y de subir la apuesta. Una cosa a tener en cuenta es que p+v+s=1.

Esta tabla recoge la probabilidad triple de acción para cada combinación de cartas del oponente. Estas probabilidades serán resultado de los cálculos del algoritmo.

La tabla tendrá un aspecto como este para cada combinación de cartas:

|  |  |  |  |
| --- | --- | --- | --- |
|  | %Pasar | %Ver | %Subir |
| ATAP | p | v | S |

Dado que estos valores son probabilidades de acción, estos valores cambiarán con el paso de las rondas.

Previamente se ha mencionado que los valores de la tabla de pesos relativos se actualizarían con las acciones del oponente y los valores de la tabla de probabilidad de acción del oponente. En función de la acción del oponente, se multiplica el valor de la tabla de la tabla de pesos relativos para esa combinación por el valor de la acción elegida de la tabla de probabilidad de acción del oponente que tenga esa mano.

Pongamos un ejemplo:

Si tenemos los siguientes pesos en la tabla de pesos relativos:

Pr {10♣ 07♠} = 0,7

Pr {09♦ 03♥} = 0,3

Pr {A♦ Q♣} = 0,64

Y las siguientes probabilidades triples de acción:

PtA {10♣ 07♠} = {0,24; 0,61; 0,15}

PtA {09♦ 03♥} = {0,61; 0,3; 0,09}

PtA {A♦ Q♣} = {0,03; 0,28; 0,69}

Si el oponente ve la apuesta, los nuevos valores de los pesos en la tabla serían:

Pr {10♣ 07♠} = 0,7\*0,61=0,427

Pr {09♦ 03♥} = 0,3\*0,3 = 0,09

Pr {A♦ Q♣} = 0,64\*0,28 = 0,1792

Como podemos observar, en caso de ver la apuesta, la combinación más probable de las tres consideradas sería {10♣ 07♠} y la menos probable sería {09♦ 03♥}. También podemos ver que la probabilidad de que sea {A♦ Q♣} también ha bajado, pues con esa mano lo más probable hubiera sido subir la apuesta.

Una vez explicado estas bases, pasamos a explicar el funcionamiento del algoritmo.

Para empezar, se tiene considerar que la información conocida en las fases varía:

* Durante el preflop, únicamente tenemos nuestra mano
* Durante el flop tenemos nuestra mano y 3 cartas comunes
* Durante el turn, tenemos nuestra mano y 4 cartas comunes
* Durante el river, tenemos nuestra mano y las 5 cartas comunes.

Por tanto, el comportamiento del algoritmo lo tenemos que adaptar para que pueda reflejar esta variación de información.

Para ello, vamos a dividir el algoritmo en tres “partes”, cada una con un funcionamiento particular:

1. Cuando no se tiene información de las cartas de la mesa. (Preflop)
2. Cuando hay información de cartas de la mesa y aún quedan cartas por salir (Flop y Turn)
3. Cuando ya está toda la información de las cartas de la mesa (River)

**Preflop**

En esta fase, solo disponemos de 2 cartas de información conocida, por lo tanto restan 50 cartas posibles, de las cuales cualquier combinación de esas 2 cartas podrían ser las que quedan en la mano del oponente, lo que significa un total de (50 2) = 1225 combinaciones posibles.

Otra de las peculiaridades de esta fase es que, no ha habido ningún tipo de apuesta por parte del oponente al comienzo de esta apuesta, por lo que los pesos son iguales uno a otro.

Como ya hemos hablado previamente, la fuerza de una mano en el preflop la hemos determinado mediante la fórmula de Chen. Es cierto que esta fórmula tiene sus limitaciones, pero como un indicador base nos sirve de guía de cara a automatizar este proceso.

Tal y como se mencionó en su momento, Chen, según su experiencia, nos indicaba varias puntuaciones por las que se podría empezar a jugar una mano. Pero, en una partida de dos jugadores, no hay posiciones, ya que no hay jugadores intermedios entre un jugador y el Dealer. Pero partiendo de ese criterio, y del cruce con Sklansky-Malmuth, podemos determinar qué manos ver, qué manos subir y qué manos desechar.

Como hemos mencionado anteriormente, una de las cualidades que buscamos en el algoritmo es que sea imprevisible, por lo que vamos a añadir porcentajes de acción, para que, aunque tengamos una acción predominante, no siempre sea preciso.

También vamos a tener en consideración la guía de <http://www.thepokerbank.com/strategy/basic/starting-hand-selection/chen-formula/> para esta consideración.

En función del valor de la fórmula de Chen, establecemos esta valoración de triples probabilidades:

|  |  |  |  |
| --- | --- | --- | --- |
| Puntuación Chen | %pasar | %ver | %subir |
| 12-20 | 0 | 0,1 | 0,9 |
| 10-11 | 0 | 0,25 | 0,75 |
| 5\*-9 | 0,25 | 0,65 | 0,10 |
| 3,5-5\*\* | 0,69 | 0,30 | 0,01 |
| Menor de 3,5 | 0,99 | 0,01 | 0 |

Para esta valoraciones y asignar estos pesos, he tomado en consideración los siguientes aspectos:

* Los pertenecientes al Grupo uno de Sklansky-Malmuth (12-20) son manos con las que son convenientes tomar la estrategia de subir la mano, por tanto, son manos que siempre se van a jugar. Además, son manos que, ante una subida, también son partidarias de resubir dicha subida.
* Los pertenecientes al Grupo 2 (10-11) son manos con las que son convenientes tomar una estrategia agresiva pero no tanto como las del grupo anterior, por tanto, son manos que prácticamente siempre se van a jugar. Además, son manos que, ante una subida, también son partidarias de ver dicha subida, pero no de subirlas
* Las manos pertenecientes a los grupos 3 al 6 (5\*-9) son las manos que, según Chen, son manos jugables según la posición que se tenga. Por tanto, podemos asumir que son manos que suelen ver las apuestas, aunque ante resubidas, suelen pasar. \*Las manos de valor 5 incluidas aquí son únicamente las parejas (si bien la pareja de 5 es la única combinación de valor 5 según la fórmula de Chen que pertenece al grupo 6, considero que, dado que las otras parejas se pueden incluir aquí ya que son una jugada hecha y, además, al solo haber un oponente, no son tan fácil de superar como si hubiera más jugadores puesto que, a más jugadores, más posibilidades de que se forme una pareja con cualquiera de las cartas comunes.
* Las manos pertenecientes a los grupos 7 y 8 (3,5-5\*\*) son manos que son difíciles de ganar, pero aún se consideran con posibilidades según la clasificación de Sklansky-Malmuth. Tanto Chen como thepokerbank recomiendan que estas manos sean descartadas, por eso se le da un mayor peso a pasar que a ver. \*\*Aquí no están incluidas las incluidas en el punto anterior, pero si están incluidas las combinaciones de AX de distinto palos siendo X>9 puesto que, aunque la fórmula de Chen les dé una valoración de 5, Malmuth-Sklansky las descarta como manos jugables. He decidido incluirlas por un razonamiento similar al anterior punto: al haber un solo oponente esta combinación de cartas gana fuerza.
* Manos por debajo de 3,5: son las combinaciones que están fuera de los grupos Sklansky-Malmuth, con la excepción mencionada en el punto anterior, por lo que son manos con muchas dificultades para ganar la jugada. Por ende, son manos prácticamente descartables.

De estos razonamientos podemos establecer una segunda tabla de probabilidad triple, ante la resubida en caso de que el oponente suba:

|  |  |  |  |
| --- | --- | --- | --- |
| Puntuación Chen | %pasar | %ver | %subir |
| 12-20 | 0 | 0,2 | 0,8 |
| 10-11 | 0 | 0,8 | 0,2 |
| 5\*-9 | 0,85 | 0,1 | 0,05 |
| 5 o menor | 0,99 | 0,01 | 0 |

En el caso de que haya una subida, el algoritmo también tomará en consideración los valores de la segunda tabla de probabilidad triple para actualizar la tabla de pesos relativos.

Pongamos un ejemplo:

Tomando en cuenta estas posibles manos: {10♠ 09♠}, {K♥ K♣}, {09♦ 03♥} y {A♦ Q♣}, teniendo las cuatro un Pr=1, y siendo sus valores de la fórmula de Chen 6, 16, -0,5 y 9 respectivamente

En el caso de que primero haya una subida, su Pr se quedaría asi:

Pr {10♠ 09♠} = 1\*0,1=0,1

Pr { K♥ K♣} = 1\*0,9= 0,9

Pr {09♦ 03♥} = 1\*0,01=0,01

Pr {K♦ Q♣} = 1\*0,75= 0,75

Y, suponiendo que el jugador resube y el algoritmo decide ver la subida, los pesos relativos se quedarían así:

Pr {10♠ 09♠} = 0,1\*0,1=0,01

Pr { K♥ K♣} = 0,9\*0,2=0,18

Pr {09♦ 03♥} = 0,01\*0,01=0,0001

Pr {K♦ Q♣} = 0,75\*0,8=0,6

Podemos observar que se van perfilando las manos del oponente según sus acciones.

Una vez actualizados los Pesos relativos, pasamos a la siguiente fase

**Flop y Turn**

He agrupado ambas fases en este punto porque tienen una composición parecida: hay varias cartas comunes, pero aún quedan cartas por salir. Además, al haber al menos 5 cartas entre la mano y la mesa, la fórmula de Chen deja de ser efectiva pues ya hay jugadas. Ahora las jugadas siguen la puntuación mencionada previamente (0 a 9, siendo 0 carta más alta y 9 Escalera Real).

Antes de nada, lo primero que hay que hacer es actualizar la tabla de probabilidades con las cartas reveladas en la fase correspondiente (3 cartas en Flop y 1 en Turn), haciendo que todas las combinaciones de cartas que incluyan dichas cartas tengan probabilidad 0.

Despues, podemos empezar a valorar la situación con la información que tenemos.

En este caso, no podemos hacer un simple cálculo para determinar qué puede tener el oponente, puesto que hay muchas más variables posibles en lo que se refiere a las posibles combinaciones de mano.

Para poder automatizar esto, tomaremos como base las ideas plasmadas en Algorithms and Assessment in Computer Poker, la tesis doctoral de Darse Billings de la universidad de Alberta. En la tesis, habla de distintas estrategias para cómo automatizar el juego de póker.

En esa tesis, se define el valor de Fuerza de Mano (FM), que es la probabilidad de que una mano sea mejor que la del oponente, al igual que el Valor Potencial de la Mano (VPM), que es la fuerza potencial con las posteriores cartas.

Estas definiciones nos sirven de base para este proyecto, aunque vamos a realizar modificaciones sobre eso para que se amolden a nuestro algoritmo.

Para empezar, vamos a calcular cómo de buena es la mano con respecto a las posibles manos del oponente. Pero antes de eso, es necesario hacer una simplificación, de cara a ayudar a los cálculos: únicamente se van a tomar en consideración variaciones en la puntuación directa de la jugada, no en su desempate. Es decir, si cada jugador tiene una pareja, se van a considerar como “jugadas iguales”, aunque en la práctica esto no tenga que ser estrictamente cierto.

Con esto, podemos empezar a ver cómo de buena es esta mano, para eso comprobamos si, con cada combinación de dos cartas posibles del oponente su jugada es mejor, peor o igual que la del jugador.

CalcularFuerzaMano(valorjugadaJugador, Mesa)

{

Float NValores superiores= NValores iguales= NValores inferiores=0;

\*Considerando cada una de las posibles manos del oponente\*

para cada (ManoOponente):

{

Jugada\_oponente = ManoOponente + Mesa;

valorjugadaOponente = calcularValorJugada(Jugada\_oponente);

si valorjugadaOponente > valorjugadaJugador -> NValores superiores= NValores superiores +1\* Pr {ManoOponente };

si valorjugadaOponente == valorjugadaJugador -> NValores iguales= NValores iguales +1\* Pr { ManoOponente };

si valorjugadaOponente < valorjugadaJugador -> NValores inferiores= NValores inferiores +1\* Pr { ManoOponente };

}

FMsuperior=(NValores inferioes)/( NValores superiores+ NValores iguales+ NValores inferiores)

FMigual=(NValores iguales)/( NValores superiores+ NValores iguales+ NValores inferiores)

FMinferior=(NValores supeiores)/( NValores superiores+ NValores iguales+ NValores inferiores)

}

Siendo FMsuperior el porcentaje en que nuestra mano es superior a la del oponente, FMigual el porcentaje en que nuestra mano es igual a la del oponente y FMinferior el porcentaje en que nuestra mano es inferior a la del oponente. Como se puede observar, no estamos únicamente contando, sino que estamos sumando el Pr de esa mano para hacer el conteo, porque no todas manos tienen la misma probabilidad de aparecer.

Pongamos un ejemplo sobre esto:

Teniendo una mano de {9♦ 8♦} y en la mesa se han revelado {Q♦ 10♦ J♦}, lo cual nos da una escalera de color. La única combinación posible con estas cartas para que nuestro oponente nos gane el oponente es con una mano de {A♦ K♦} y no habiendo ninguna posible combinación de manos que iguale la nuestra.

Durante el preflop, el oponente ha visto la mano, sin hacer ninguna subida, por lo Pr {A♦ K♦} = 1\*0,2=0,2. Lo cual, hace que sea menos probable que el oponente tenga dicha mano que si hubiera tenido una jugada agresiva en el preflop, haciendo que el valor de FMinferior sea menor.

En estas fases, este valor FM no es suficiente para perfilar una estrategia, pues aún quedan cartas por salir e información que revelarse: en el Flop 2 cartas y en el Turn 1 carta.

Para esto, hay que intentar suponer cómo evolucionarán las manos con las posibles cartas. Hay dos maneras de enfocarlo:

* Durante el flop considerar ambas posibles cartas Considerar únicamente la siguiente carta a salir.

En el primer caso, se tendría que considerar, para cada una de las posibles manos del oponente, cada una de las posibles 45 cartas restantes del mazo y, así mismo, para cada una de esas 45 cartas restantes, cómo afectaría la jugada para cada una de las 43 cartas restantes durante el flop y luego durante el turn considerar de nuevo las 43 cartas restantes para cada mano posible del oponente, aunque solo habría que calcularlo una vez

Mientras que, en el segundo caso, únicamente se tiene que considerar para cada una de las posibles manos del oponente, cómo afectaría a la jugada la aparición de cada una de las posibles 45 cartas restantes del mazo durante el flop y para el turn considerar para cada mano posible del oponente la variación de la jugada con las 43 cartas restantes del mazo. Es decir, hacer dos cálculos uno para cada fase.

Vamos a utilizar la segunda opción, principalmente por sencillez de computación. Si bien es cierto que se pierde precisión a la hora de estimar, se gana bastante en rendimiento ya que son menos combinaciones a tener en cuenta.

Para ello, calculamos el Valor Potencial de la Mano (VPM), de manera similar a cómo lo calcula Darse Billings en su tesis. Es decir, para cada mano del oponente, analizamos si partimos de estar en ventaja, en igualdad o en inferioridad y analizamos cómo esa situación mejora, empeora o se queda igual (dando lugar a un total de 9 posibilidades)

CalcularPotencialMano(ManoJugador, Mesa, valorjugadaJugador)

{

Float array VPT[3] ={0, 0, 0}; //Que servirá para contar el total

Float array VP[3][3] ={0, 0, 0; 0, 0 ,0; 0, 0, 0};//Que servirá para contar cada una de las 9 //posibilidades.

int i=-1;

Para cada (ManoOponente)

{

Jugada\_oponente = ManoOponente + Mesa;

valorjugadaOponente = calcularValorJugada(Jugada\_Oponente);

si valorjugadaOponente > valorjugadaJugador -> i=0; //Situación en que tenemos peor //mano que el oponente

si valorjugadaOponente == valorjugadaJugador -> i=1; //Situación en que tenemos //misma jugada que el oponente

si valorjugadaOponente < valorjugadaJugador -> i=2; //Situación en que tenemos mejor //mano que el oponente.

VPT[i]=VPT[i]+ Pr { ManoOponente };

Para cada(CartaRestante):

{

Jugada\_Jugador’=ManoJugador+Mesa+CartaRestante;

Jugada\_Oponente’=ManoOponente+Mesa+CartaRestante;

valorjugadaOponente’= calcularValorJugada(Jugada\_Oponente’);

valorjugadaJugador’= calcularValorJugada(Jugada\_Jugador’);

si (valorjugadaOponente’> valorjugadaJugador’) ->VP[i][0]= VP[i][0]+1\* Pr {ManoOponente}; //Peor jugada con la nueva carta

si (valorjugadaOponente’==valorjugadaJugador’) ->VP[i][1]= VP[i][1]+1\* Pr {ManoOponente}; // Misma jugada con la nueva carta

si (valorjugadaOponente’< valorjugadaJugador’) ->VP[i][2]= VP[i][2]+1\* Pr {ManoOponente}; // Mejor jugada con la nueva carta

}

}

//Ahora calculamos los potenciales de la mano, tanto positivo (que, yendo por detrás, mejora la //mano) como negativo (Que, yendo por delante, la mano del oponente mejora y nos supera)

PotPositivo = VP[0][2]/VPT[0];

PotNegativo = VP[2][0]/VPT[2];

}

Ahora tenemos tanto la fuerza de la mano como la posible variación con la siguiente carta. Con esto, podríamos establecer ya una estrategia, pero hay más factores a tener en cuenta con respecto a esto.

Lo primero, el concepto de Odds que ya hemos tratado anteriormente, como un factor más que es el factor de farol/bluff.

Lo primero, vamos a calcular los Odds de mano, es decir, cúantas cartas nos mejoran la jugada, es decir, cuántas de las posibles cartas restantes nos mejoran la mano. Para ello, vamos a valorar cada una de las cartas que desconocemos (considerando también las dos cartas del oponente.

CalculoOddsMano(ManoJugador, Mesa, valorjugadaJugador)

{int cartamejora=0;

Para cada(CartaRestante):

{

Jugada\_Jugador’=ManoJugador+Mesa+CartaRestante;

valorjugadaJugador’= calcularValorJugada(Jugada\_Jugador’);

Si valorjugadaJugador’ > valorjugadaJugador ->cartamejora++;

}

OddMano=cartamejora/(cartasMazo-cartamejora)

}

Después, vamos a calcular el Odd del bote, es decir, cual es la relación entre cuánto dinero hay en el bote y cuanto tenemos que apostar. En otras palabras, el inverso de cuánto dinero ganaríamos por cada € que apostemos.

CalculoOddsBote(ApuestaActualJugador, ApuestaActualOponente)

{

Diferencia = ApuestaActualOponente-ApuestaActualJugador;

Bote = ApuestaActualJugador + ApuestaActualOponente + Diferencia;

OddBote=Diferencia/Bote

}

Con estos factores, ya tenemos datos suficientes para calcular una probabilidad triple de acción para el algoritmo.

Teniendo en cuenta que p+v+s=1, la fórma más sencilla de conseguir que se cumpla esto siempre es que uno de los tres valores sea una combinación lineal de los otros dos. Debido a que la opción que se considera como “pasiva” es la opción de Ver, vamos a reformular esa ecuación como v=1-(p+s). Es decir, que v será siempre una función de los otros dos.

Además, tenemos la relación por la que se debe apostar o no una mano en función de los Odds: el odd de bote debe ser menor que el odd de mano para considerarlo una buena mano. Por este motivo, va a sumar para la probabilidad de Subir en el caso de que OddBote<OddMano o restar en el caso inverso, mientras que tendrá el comportamiento opuesto de cara a la probabilidad de pasar la mano.

De esta manera podemos sacar las probabilidades de Subir y de pasar una apuesta, y, por ende, de ver la mano.

s= FMsuperior+(1- FMsuperior)\*PotPositivo+(OddMano-OddBote)

p= FMinferior+(1- FMinferior)\*PotNegativo+(Oddbote-OddMano)

v=1-(p+s)=1-(FMsuperior+(1-FMsuperior)\*PotPositivo+(OddMano-OddBote)+FMinferior+(1-FMinferior)\* PotNegativo+(Oddbote-OddMano))=1-(FMsuperior+(1-FMsuperior)\*PotPositivo+FMinferior+(1- FMinferior)\*PotNegativo)

Antes hemos mencionado un factor que se ha mencionado que influenciaba en estas probabilidades, pero no se ha tomado en cuenta hasta ahora y es el factor de farol. Como se mencionó previamente, los faroles son una estrategia para apostar sin tener una mano ganadora, ya sea cuando no tenemos posibilidad de ganar una mano (farol puro) o cuando no sabemos si nuestra mano puede llegar a ser una mano ganadora.

Para representar este factor, así como para mantener la imprevisibilidad del algoritmo, vamos a caracterizar este comportamiento mediante un número aleatorio. Dado que son manos donde las probabilidades están en nuestra contra, el algoritmo solo va a considerar la posibilidad de “echarse” un farol cuando p sea la mayor de las tres probabilidades, pero no siempre que vaya perdiendo será partidario de seguir esta estrategia, pues acaba perdiendo fuerza.

Por eso determinaremos este comportamiento determinando si se hace el farol con un numero aleatorio entre 0 y 100, y si se cumple que Random≥p\*100, entonces se hará el farol.

En este caso, v=v+p/2, s=s+p/2 y p=0.

Ahora tenemos que rellenar la Tabla de Probabilidad de acción del oponente. La manera ideal sería repetir el proceso que hemos seguido para el algoritmo con cada combinación posible de manos. Pero vamos a hacerlo de manera más sencilla para simplificar la computación.

Para ello, vamos a aplicar una modificación de la función de CalculoFuerzaMano, creando la función CalculoProbabilidadAccion.

CalculoProbabilidadAccion(Mesa)

{

Float NValores superiores= NValores iguales= NValores inferiores=0;

\*Considerando cada una de las posibles manos del oponente\*

para cada (ManoOponente):

{

Jugada\_oponente = ManoOponente + Mesa;

valorjugadaOponente = calcularValorJugada(Jugada\_oponente);

Para cada (ManoPosibleJugador)

{

Jugada\_Jugador = ManoPosibleJugador + Mesa;

valorjugadaJugada = calcularValorJugada(Jugada\_oponente);

si valorjugadaOponente < valorjugadaJugador -> NValores superiores= NValores superiores +1;

si valorjugadaOponente == valorjugadaJugador -> NValores iguales= NValores iguales +1;

si valorjugadaOponente >valorjugadaJugador -> NValores inferiores= NValores inferiores +1};

}

s[ManoOponente] =(NValores inferioes)/( NValores superiores+ NValores iguales+ NValores inferiores)

v[ManoOponente] =(NValores iguales)/( NValores superiores+ NValores iguales+ NValores inferiores)

p[ManoOponente]=(NValores supeiores)/( NValores superiores+ NValores iguales+ NValores inferiores)

}

}

Este proceso (actualizar pesos, calcular factores, calcular probabilidad triple del algoritmo y rellenar tabla de probabilidades de acción del oponente) se repite con cada acción de la ronda de apuestas durante Flop y Turn.

**River**

Durante el River, las manos ya no pueden variar, pues se han revelado las 5 cartas de la Mesa. Por esta razón, los Odds ni el potencial de las manos tienen sentido tenerlos en consideración, pues las manos y jugadas son las que son.

En esta fase, el cálculo se simplifica bastante y se parece bastante a cómo se ha calculado la probabilidad triple de acción de cada mano del oponente.

CalcularProbabilidadTriple(valorjugadaJugador, Mesa)

{

Float NValores superiores= NValores iguales= NValores inferiores=0;

\*Considerando cada una de las posibles manos del oponente\*

para cada (ManoOponente):

{

Jugada\_oponente = ManoOponente + Mesa;

valorjugadaOponente = calcularValorJugada(Jugada\_oponente);

si valorjugadaOponente > valorjugadaJugador -> NValores superiores= NValores superiores +1\* Pr {ManoOponente };

si valorjugadaOponente == valorjugadaJugador -> NValores iguales= NValores iguales +1\* Pr { ManoOponente };

si valorjugadaOponente < valorjugadaJugador -> NValores inferiores= NValores inferiores +1\* Pr { ManoOponente };

}

s=(NValores inferioes)/( NValores superiores+ NValores iguales+ NValores inferiores)

v=(NValores iguales)/( NValores superiores+ NValores iguales+ NValores inferiores)

p=(NValores supeiores)/( NValores superiores+ NValores iguales+ NValores inferiores)

}

De esta manera, el factor Fuerza de Mano sería el único que nos queda válido de los considerados arriba. Aun así, el factor de Farol sigue aplicando aquí, funcionando de la misma manera que en Flop y en Turn.

El cálculo de la probabilidad triple de acción del oponente sigue siguiendo el mismo proceso que en Flop y en Turn.