

UNIVERSITAT POLITÈCNICA DE VALÈNCIA

DEPARTAMENTO DE SISTEMAS INFORMÁTICOS Y COMPUTACIÓN



MÁSTER EN INTELIGENCIA ARTIFICIAL, IMAGEN DIGITAL Y
RECONOCIMIENTO DE FORMAS

TRABAJO FIN DE MÁSTER

CURSO 2016/2017

**APLICACIÓN DE TÉCNICAS INTELIGENTES PARA
LA RESOLUCIÓN DE PROBLEMAS**

José María Aceituno Peinado

Septiembre 2017

Tutor: Federico Barber

Índice

1. Introducción	4
1.1 Teoría de juegos.....	4
1.1.1 Introducción a Teoría de juegos.....	4
1.1.2 Tipologías de juegos.....	8
1.2 Teoría de la decisión.....	15
2. Planteamiento del TFM	18
2.1 Motivación.....	18
2.2 Objetivos.....	20
2.3 Metodología. Plan de desarrollo.....	21
2.4 Resumen de los puntos de la memoria.....	22
2.5 Conclusiones previas.....	23
3. Desarrollo del TFM	24
3.1 Juego del Poker.....	24
3.1.1 Definición.....	24
3.1.2 Historia.....	25
3.1.3 Variantes	26
3.1.4 Reglas.....	27
3.2 Relación del Poker con Teoría de Juegos.....	33
3.3 Estado del arte en el estudio científico del Poker.....	37
3.3.1 Introducción.....	37
3.3.2 Sistemas desarrollados.....	39
3.4 Propuesta de algoritmo.....	45
3.4.1 Aproximación inicial.....	45
3.4.2 Información a considerar.....	46
3.4.3. Estrategias del juego.....	47
3.4.4 Propuesta de aprendizaje.....	48
4. Diseño del algoritmo	49
4.1 Introducción general.....	49
4.2 Jugar partida.....	52
4.2.1 Parametrización de los datos.....	53
4.2.2 Elección de alternativa. Toma de decisión.....	62

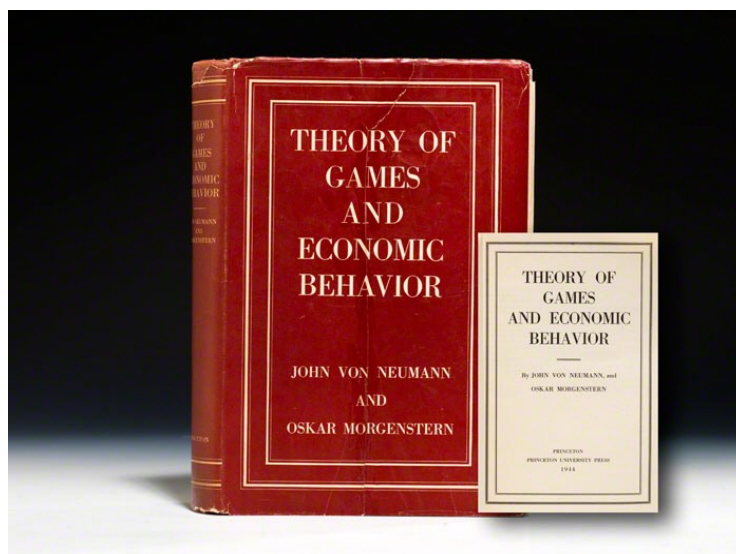
4.3	Aprendizaje.....	69
4.3.1	Introducción.....	69
4.3.2	Recopilación de datos.....	71
4.3.3	Evaluación de datos recopilados.....	72
5.	Implementación y codificación	75
6.	Evaluación, pruebas y resultados	78
6.1	Introducción.....	78
6.2	Entrenamiento.....	79
6.3	Test. Jugador Bot vs Jugador humano.....	86
7.	Conclusiones Finales	88
8.	Bibliografía	89

1 Introducción

1.1 Teoría de Juegos

1.1.2 Introducción a Teoría de Juegos

Nuestro proyecto se enmarca dentro de la ciencia de “Teoría de Juegos”. El concepto de teoría de juegos se inició con Von Neumann y Morgenstem en la publicación de un libro en 1944 llamado “The Theory of Games and Economic Behaviour”.



(Imagen 1) Libro de John Von Neumann y Oskar Morgenstem “Teoría de juegos y comportamiento económico”, 1944

Hay que reconocer que economistas como Cournot o Edgeworth fueron vanguardistas en este campo anteriormente al anterior libro, es decir, ya en el siglo XIX habían anticipado algunas ideas. [1]

También es de destacar en este campo contribuciones de principios del siglo XX hechas por ilustres científicos como el matemático y político francés Émile Borel que publicó varias investigaciones sobre Teoría de juegos, o el berlinés Ernst Friedrich Ferdinand Zermelo que fue un matemático y lógico alemán que aportó bastante a la materia como sus estudios de “Problemas de Teoría de conjuntos”, los cuales se enmarcan dentro de los “Problemas de Hilbert”, que son una lista de 23 problemas no resueltos expresados en una conferencia

internacional del Matemáticos en París en el año 1900.

Von Neumann ya adelantó algunos conceptos en 1928 en artículos publicados, pero en cualquier caso, hay cierto consenso científico al afirmar que fue el libro de “ The Theory of Games and Economic Behaviour” el que estableció el concepto de “Teoría de Juegos” tal y como lo entendemos hoy en día.

Definiciones de Teoría de Juegos:

Es una colección de conceptos y modelos que versan sobre el comportamiento humano en contextos estratégicos en los que un jugador racional depende de los supuestos que él mismo haga respecto del posible comportamiento de otros jugadores. [2]

Es una rama de las matemáticas y de la economía que estudia la elección de la conducta óptima de un individuo cuando los costes y los beneficios de cada opción no están fijados de antemano, sino que dependen de las elecciones de otros individuos. [3]

Es la rama de las matemáticas que concierne al análisis de estrategias para tratar con situaciones competitivas donde el resultado de la elección de un participante depende críticamente de las acciones de los otros participantes. La teoría de juegos ha sido aplicada a contextos como guerra, negocios o biología. [4]

Como podemos observar hay varias definiciones para Teoría de juegos pero hay una idea que más o menos subyace en todas las definiciones.

La teoría de juegos tiene un objetivo general, abarca cuestiones que son básicas para todas las ciencias sociales, Esto puede nos ofrece una visión particular en situaciones de índole política, social o económica que repercute en individuos que tienen diferentes metas o preferencias. [5]

Un ejemplo clásico de Teoría de juegos es el **dilema del prisionero**:

Dos personas son arrestadas, encarceladas y se les fija la fecha del juicio. A priori no hay pruebas contundentes contra ellos. El fiscal del caso habla con cada prisionero por separado y les presenta una oferta:

Si el prisionero A confiesa contra su socio, el prisionero B, todos los cargos en su contra serán retirados y la confesión será usada como evidencia para condenar al otro, que recibirá una sentencia de 10 años.

Si el prisionero A no confiesa contra B, pero el prisionero B sí confiesa contra el prisionero A, entonces A será condenado a 10 años y su socio quedará libre.

Si ambos confiesan, serán condenados a 5 años de prisión.

Si ninguno confiesa, serán condenados solo a 6 meses de prisión por delitos menores.

Podemos apreciar la matriz de pagos en la imagen (2)

Desde el punto de vista racional, confesar sería la mejor opción pero si ambos confiesan el castigo es peor que si ambos callan. [6]

	Prisionero B Coopera	Prisionero B Traiciona
Prisionero A Coopera	Cada uno sirve 6 meses	Prisionero A: 10 años Prisionero B: Sale libre
Prisionero A Traiciona	Prisionero A: Sale libre Prisionero B: 10 años	Cada uno sirve 5 años

(Imagen 2) Tabla resultado del dilema del prisionero

En nuestro TFM vamos a tratar el juego del Poker Texas Hold'em, pero sería conveniente empezar definiendo técnicamente lo que entendemos por juego.

Definiciones de Juego:

Actividad recreativa física o mental en la que compiten dos o más personas sometiéndose a unas reglas.[7]

Un juego es una situación en la que compiten dos o más jugadores. [8]

Un juego es una situación en la que los individuos deben tomar decisiones estratégicas y en la que el resultado final depende de lo que cada uno decida hacer. [9]

Cualquier problema de toma de decisiones, donde el rendimiento que obtiene una persona depende no solo de sus propias decisiones sino también de las decisiones de las otras personas que participan en el juego. [10]

Cuando hablamos de juegos debemos definir los siguientes conceptos:

Jugador o Agente:

es el ente que toma las decisiones.

Acción:

las acciones son las diversas opciones que el jugador puede elegir. Los jugadores no tienen por qué tener las mismas acciones para escoger.

Función de Matriz de pagos:

nos da la utilidad por cada jugador para todas las combinaciones de acciones que tomen los otros jugadores.

	Jugador A: escoge 1	Jugador A: escoge 2
Jugador B: escoge 1	A: +2, B: -2	A: -3, B: +3
Jugador B: escoge 2	A: -3, B: +3	A: +4, B: -4

Ejemplo de matriz de pagos para el juego “pares o nones”

1.1.2 Tipología de juegos

Hay diversas tipología de los juegos que van en función de su estructura y características propias. Vamos a definir las diversas tipologías y señalar cuales son las que nos repercute en nuestro juego elegido, Poker Texas Hold'em.

Según el número de Jugadores:

- Juegos de un solo jugador
- Juegos de dos jugadores
- Juegos de múltiples jugadores

Según el número de estrategias:

para categorizar los juegos en función de sus estrategias, valoramos las alternativas de decisión de cada jugador, las cuales pueden ser:

- Finitas alternativas
- Infinitas alternativas

Según la relación entre los jugadores:

- Juegos sin coaliciones, donde no se pueden hacer ni confederaciones ni uniones.
- Juegos cooperativos, donde las coaliciones se firman con anterioridad.
- Juegos con coaliciones ocultas.

Según el tipo de pago:

- Juegos de suma-constante, donde la suma de los beneficios de todos los jugadores siempre es la misma. El beneficio de cada jugado se resta del contrario o contrarios.

- Juegos de suma-cero (o suma nula): Caso particular de suma-constante, donde en cada caso la suma de beneficios es 0. En caso de dos jugadores, la ganancia o pérdida de un participante es exactamente la pérdida o ganancia del oponente.

-Juegos de suma no-nula, o juegos de suma no-constante, típicos empresariales.

Según la información disponible:

- Juegos de información perfecta donde un jugador conoce todas las opciones del resto de jugadores y sus recompensas. Un ejemplo es el ajedrez.

- Juegos de información imperfecta (o incompleta) cuando las decisiones de los jugadores se toman simultáneamente y no es posible conocer ni preveer las decisiones que el resto de jugadores pueden tomar. Un ejemplo de información imperfecta es casi cualquier juego de cartas donde haya cartas escondidas.

Según la transferencia de información:

- Juegos competitivos, donde los jugadores compiten por un único bien, uno gana y el otro pierde. Si es suma nula entonces se llama “estrictamente competitivo”-

- Juegos cooperativos, donde se transfiere información, se coopera, se persigue el bien común.

Según la aleatoriedad:

- Juegos deterministas, donde las consecuencias dependen de las decisiones de los jugadores, sin incertidumbre o aleatoriedad.
- Juegos no deterministas, donde los resultados no están completamente determinados. Y dentro de esta categoría destacamos la teoría de juegos frente a la naturaleza, en la que existe aleatoriedad, los resultados dependen de las acciones de otros jugadores y del azar.

Según Número de movimientos:

- Juegos simultáneos, o estáticos donde se juega a la vez, sin conocer la elección del contrario.
Ejemplo: piedra, papel y tijera.
- Juegos de un paso, donde cada jugador tiene su turno, y hace su movimiento.
- Juegos multipasos, cada jugador puede realizar más de un movimiento a la vez.
- Juegos reiterativos, juegos simultáneos jugados repetidas veces.



(Imagen 3) "Piedra, papel o tijera" es un ejemplo de juego simultáneo.

Estrategias Puras y Mixtas

En teoría de juegos, cuando un jugador puede elegir entre un conjunto de estrategias posibles, hablaremos de estrategia pura cuando la probabilidad de que escojamos una estrategia determinada es 1. (y por tanto probabilidad 0 para el resto de estrategias)

Por otro lado, hablaremos de estrategias mixtas cuando no haya ninguna estrategia con probabilidad 1, sino que varias estrategias tengan asignadas una probabilidad entre 0 y 1. Es decir, hay una distribución de la probabilidad entre el conjunto de estrategias.

Estrategias Dominantes.

La teoría de estrategias dominantes dice que una estrategia e para un jugador domina en sentido fuerte a una estrategia e' , si la Utilidad de (e) es mayor a la Utilidad de (e'), es decir, Utilidad (e) $>=$ Utilidad (e'), para cada estrategia posible de los otros jugadores.

Una estrategia e es dominante si domina a todas las demás estrategias del jugador.

Estrategia Maximin y Minimax

Maximin es un término comunmente usado para juegos de suma no-nula donde se adopta una estrategia que maximiza la ganancia mínima que puede obtenerse.

	Contrario: A	Contrario: B	Contrario: C	
Yo: A?	Yo: 9, C:1	Yo: 1, C:9	Yo: 2, C:8	MAXIMIN (YO) 1
Yo: B?	Yo: 6, C:4	Yo: 5, C:5	Yo: 4, C:6	4
Yo: C?	Yo: 7, C:3	Yo: 8, C:2	Yo: 3, C:7	3
	1	2	6	← MAXIMIN (Contrario)

(Imagen 4) Matriz de pagos para encontrar el Maximin

En este ejemplo, mostramos la matriz de pagos que tendrían 2 jugadores si tuvieran que elegir entre las opciones A,B y C cada uno.

Para nosotros, entre las 3 opciones la que nos garantiza la mejor ganancia mínima es la opción B. Que nos daría una ganancia de como mínimo 4.

Para nuestro contrario, la opción que le ofrece la ganancia mínima es C.

Minimax es similar a Maximin, es una estrategia donde se pretende minimizar la máxima pérdida esperada. Recordamos que en maximin no hay pérdidas, sino ganancias mínimas.

Equilibrio de Nash

El equilibrio de Nash es una solución a un juego entre 2 o más jugadores junto con sus estrategias que maximizan sus ganancias de modo que a ninguno de los jugadores le conviene cambiar de estrategia ya que empeoraría su pago o resultado.

Se puede expresar como que en un juego con N jugadores, las estrategias $\{ S_1^*, S_2^*, S_3^*, \dots, S_n^* \}$ son un Equilibrio de Nash si, para cualquier jugador i , la estrategia S_i^* es la mejor respuesta a las estrategias $\{ S_1^*, S_2^*, S_3^*, \dots, S_{i-1}^*, S_{i+1}^*, S_n^* \}$ de los demás jugadores:

$$U(S_1^*, S_2^*, \dots, S_{i-1}^*, S_i^*, S_{i+1}^*, \dots, S_n^*) \geq U(S_1^*, S_2^*, \dots, S_{i-1}^*, S_i, S_{i+1}^*, \dots, S_n^*)$$

\forall jugador $i \in N$, y toda estrategia $S_i \in S$.

O dicho de otro modo, dados dos jugadores, $(A=a^*, B=b^*)$ es un equilibrio de Nash, si la estrategia $A=a^*$ es óptima para A frente a la estrategia $B=b^*$; y $B=b^*$ es óptima para B frente a $A=a^*$.

Pareto Óptimo

Decimos que una solución es Pareto óptima si no hay ninguna otra solución que todos los jugadores prefieran.

Otra definición es que dado un conjunto de individuos con sus perfiles de estrategia y alcanzada una solución, no hay un cambio de opción de ningún individuo que al aplicarla y mejorar su situación no empeore también la situación de los demás.

Por ejemplo, en el dilema del prisionero explicado anteriormente la solución Pareto óptima sería que los dos prisioneros no se delataran, dado que el beneficio global de ambos sería el mejor, 6 meses cada uno de cárcel, que significa 1 año en total. Sin embargo esta solución no es estable. Dado que en esta situación, cualquiera de los dos prisioneros estaría tentado a delatar a su compañero y por tanto librarse de la cárcel. Por tanto, la solución de ambos de no delatar es un Pareto óptimo pero no un equilibrio de Nash.

Punto de Silla

El punto de silla es una situación en teoría de juegos donde las estrategias maximin de ambos jugadores coinciden. Es decir, a ninguno de los dos jugadores le interesa cambiar de estrategia dado que empeorarán su resultado.

Si nos fijamos en la matriz de pagos anterior, no existe punto de silla, debido a que los dos jugadores, a sabiendas de la elección del contrario, están tentados a cambiar de estrategia para obtener más ganancias.

Sin embargo, en la siguiente matriz de pagos, si ambos jugadores usan una estrategia maximin, se producirá un punto de silla dado que ninguno de los dos jugadores mejorará su ganancia cambiando de estrategia.

	Contrario: A	Contrario: B	Contrario: C	MAXIMIN (YO) ↓
Yo: A	Yo: 9, C:1	Yo: 1, C:9	Yo: 2, C:8	1
Yo: B	Yo: 6, C:4	Yo: 5, C:5	Yo: 4, C:6	4
Yo: C	Yo: 7, C:3	Yo: 8, C:2	Yo: 3, C:7	3
	1	2	6	← MAXIMIN (C)

La elección [Yo:B, Contrario:C], es un Punto de Silla

(Imagen 5) Un punto de silla conforma un caso de equilibrio de Nash.

1.2 Teoría de la Decisión

Cuando diseñamos un sistema que debe tomar decisiones óptimas para maximizar los resultados, podemos hablar de diferentes categorías:

- Decisiones deterministas:

son conocidas todas las alternativas y consecuencias de cada una sin incertidumbre.

-Decisiones en entornos no deterministas:

los resultados o consecuencias de las acciones no están completamente determinados. Y hay dos tipos:

- Decisiones con riesgo: Donde cada opción tiene una probabilidad conocida

- Decisiones con incertidumbre: Donde la probabilidad es completamente desconocida

Toma de decisiones simples

En juegos de forma normal, cuando los jugadores implementan un perfil de estrategia, ellos obtienen una utilidad o pago. Estos pagos son el beneficio (*outcome*) del juego. La idea básica es que los jugadores quieran maximizar su utilidad. Para una perfil de estrategia s , se denota con $U_i(s)$ la utilidad que el jugador i tendría en el juego si s fuese la combinación de estrategias elegidas por los jugadores.

Nos referimos a la función U_i como funciones de utilidad. Así pues el hecho de que $U_i(s) > U_i(s')$ significa que el jugador i prefiere el beneficio resultante del perfil de estrategia s antes que al perfil de estrategia s' . [11]

Las preferencias de un jugador son calculadas por su función de Utilidad que les asigna un número para expresar la deseabilidad de un estado. La utilidad esperada (EU) de una acción dada una evidencia, $EU(a|e)$ es la media de las utilidades resultantes, ponderadas por la probabilidad de que su beneficio(*outcome*) suceda:

$$EU(a|e) = \sum \text{Prob}(\text{Result}(a) = s' | a, e) U(s)$$

El principio de la máxima utilidad esperada (MEU) nos dice que un jugador racional debe elegir la acción que maximiza la utilidad esperada.

$$\text{Action} = \text{argmax}_a EU(a|e)$$

El principio de la máxima utilidad esperada es básico para la Inteligencia Artificial. Un agente o jugador inteligente tiene que calcular varias cantidades de modo que se maximice la utilidad de sus acciones.

Es importante recordar que la existencia de una función utilidad que describe las preferencias de un agente o jugador no significa que el jugador está explícitamente maximizando la función de utilidad. El comportamiento racional puede ser generado de varias formas.

Hay algunos axiomas racionales que un jugador o agente debe obedecer pero un agente o jugador puede tener las preferencias que él quiera. Por ejemplo un agente puede preferir un Automóvil Ford de 1973 en vez de un Mercedes-Benz nuevo.

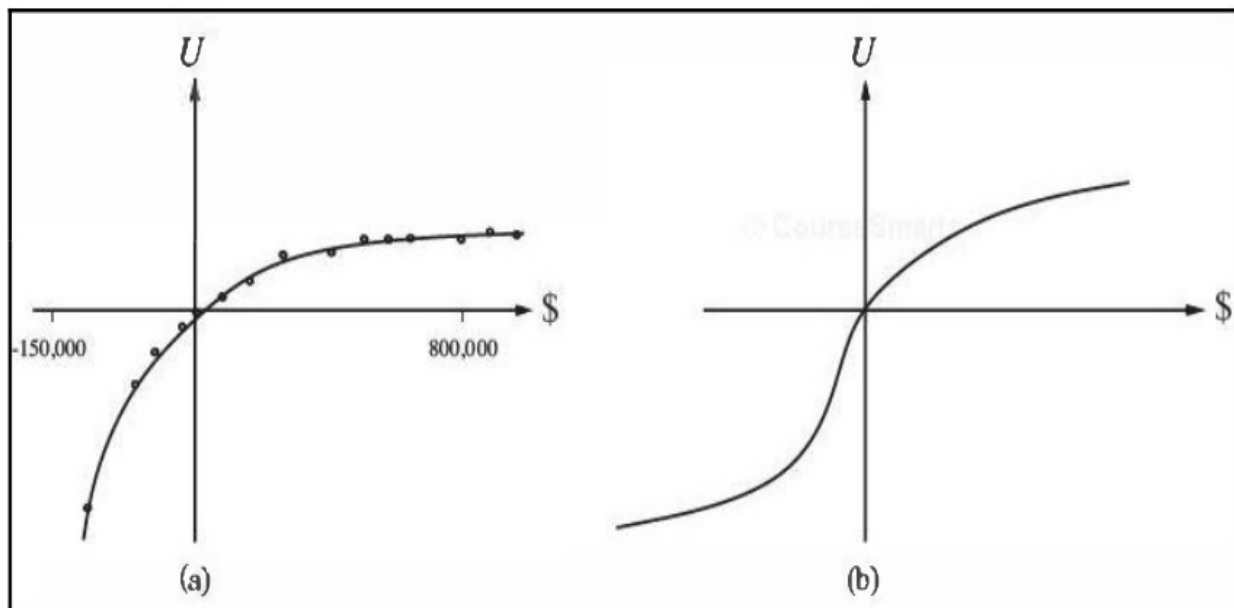
La teoría de la utilidad tiene sus inicios en la economía dado que esta nos aporta una medida obvia que es el dinero. El dinero tiene un papel muy importante en las necesidades de los humanos.

Es usual que un agente o jugador prefiera más dinero que menos, decimos que ese agente tiene una preferencia monótona por el dinero. Esto no significa que el dinero signifique su función utilidad, y para mostrarlo ponemos un ejemplo:

En un concurso de televisión un jugador obtiene 1 millón de euros, el presentador del programa le ofrece jugar ese millón de euros a cara y cruz con un moneda, de modo que si sale cara, el jugador obtiene 2'5 millones de euros, pero si sale cruz no se lleva nada. La mayoría de jugadores humanos elegirían la opción de no arriesgar y mantener ese millón de euros. Ahora bien, el valor monetario esperado del juego es:

$$1/2(0€) + 1/2 (2.500.000€) = 1.250.000 €$$

Y esto es mayor que el millón de euros original. Pero eso no significa que aceptar la apuesta sea la decisión racional. Para determinar qué hacer necesitamos asignar utilidades a los estados resultantes. La utilidad no es directamente proporcional al valor monetario, dado que la utilidad del primer millón de euros es muy alta (Imagen 6a) y la del segundo no es tan alta (Imagen 6b).



(Imagen 6) Utilidades del ejemplo expuesto sobre el concurso

2 Planteamiento del TFM

2.1 Motivación

Como alumno del Master en Inteligencia Artificial, Reconocimiento de formas e Imagen Digital por la Universidad Politécnica de Valencia, y habiéndolo cursado y aprobado todas sus asignaturas, me di cuenta que el área que más me atraía de esta ciencia es la que se centra más puramente en las técnicas de inteligencia artificial.

Habiendo cursado la asignatura de T.I.A. (Técnicas en Inteligencia Artificial) con el profesor Federico Barber, descubrí como la inteligencia artificial aplicada a la computación puede resolver variedad de problemas y juegos.

En la asignatura yo realicé un algoritmo que resolvía el problema del caballo del ajedrez.[25] Se me dio bien dado que llegué a solucionarlo, incluso con tableros grandes de 200x200. Conseguí una muy buena nota en esta asignatura y eso me motivó a seguir realizando desafíos de esa envergadura.

El juego de cartas del Poker, en su modalidad de Texas Hold'em sin límite (luego explicaremos mejor el juego) siempre me ha gustado por ser un juego donde se combina dos factores muy diferentes: la psicología humana y la probabilidad matemática.

Pensé que sería un buen acierto intentar aplicar la inteligencia artificial para este juego de cartas, Lo primero que hice fue investigar sobre si ya existían programas informáticos que realizaban esta tarea y efectivamente así era.

Investigando un poco más, me di cuenta que en la Universidad de Alberta, en Canadá hay mucho avance en esta área, incluso tienen un departamento de investigación solamente dedicado al Poker. Como comentaremos más adelante, dichos investigadores científicos han conseguido resolver el juego del Poker en su variante Texas Hold'em con límite en Head-up, es decir, con 2 jugadores. Y también han desarrollado buenos algoritmos, que no resuelven el juego, pero sí presentan jugadores artificiales muy competitivos para la variante del poker de sin límite , uno contra uno.

Sin embargo, todos estos estudios de la Universidad de Alberta y otras más que luego comentaremos, van enfocados a resolver o hallar buenos bots (software que emula un jugador) para el juego del Poker cuando es en modo Head-up, es decir, dos jugadores.

A partir de ee punto, mi desafío fue claro, quería desarrollar un algoritmo que fuese capaz de jugar a Poker en una mesa donde hay más de 2 jugadores, a diferencia de los estudios de la Universidad de Alberta y otras.

2.2 Objetivo

El objetivo de este TFM se compone de 2 fases. La primera es la creación por medio del desarrollo de software de una plataforma que permita jugar al juego de cartas *Poker Texas Hold'em Sin límite* tanto a jugadores humanos como a *jugadores bots*, con jugadores bots nos referimos a un algoritmo que emule a un jugador real y pueda participar en la plataforma. Necesitamos que dicha plataforma permita que tanto humanos como los jugadores bots puedan coexistir y competir en la misma partida de Poker.

Una vez completada la primera fase, la siguiente fase es la creación de un jugador bot por medio del desarrollo de un algoritmo que trate de jugar de manera inteligente en una partida de poker, que será jugada en la plataforma previamente creada. El bot debe ser capaz de aprender de sus errores durante el juego y modificar así su estrategia de juego para intentar tener más éxito en sus futuras partidas.

2.3 Metodología. Plan de desarrollo

Una vez está fijado el objetivo de este TFM, se debe establecer cuales serán los pasos a dar para alcanzar el objetivo propuesto.

Para lograr esto, primero se creará la plataforma donde se permitirá jugar al Poker, para ello se leerán y comprenderán todas las reglas del Poker, y diseñaremos una plataforma donde toda la normativa del *Poker Texas Hold'em Sin límite* quede implantada.

Después se crearán el algoritmos que emula a jugadores reales de Poker a través de agentes artificiales que llamaremos bots. A estos se les dotará de un criterio determinado y configurable para jugar al Poker con una estrategia inteligente y adaptativa.

Una vez creado el algoritmo, realizaremos una serie de pruebas. Estas pruebas serán de 2 tipos:

- primero se enfrentarán en la misma partida varios jugadores bots con el mismo algoritmo pero con diferentes valores en sus parámetros. El resultado de esta prueba clarificará qué configuración de parámetros es la que daa mejores resultados.

- Una vez tengamos la configuración más exitosa del algoritmo aplicada a los jugadores bots, enfrentaremos en una mesa a dicho jugador bot contra un jugador humano.

Los resultados que se obtengan serán analizados y a partir de ellos se realizarán unas conclusiones.

2.4 Resumen de los puntos de la memoria

La memoria refleja el alcance del proyecto realizado. La configuración de dicha memoria se desarrollará mediante las siguientes fases de trabajo:

- Se definirá en primer lugar el juego del Poker, así como su historia, sus variantes y sus peculiaridades.
- Se establecerá la relación del Poker con Teoría de Juegos
- Se comentarán los estudios más relevantes en el mundo científico respecto a esta área
- Se establecerá la propuesta propia para enfrentar el problema
- Se establecerá un diseño concreto del algoritmo. Se desarrollará
- Se implementará y codificará en los lenguajes y con unas herramientas predeterminadas.
- Se realizarán las pruebas pertinentes, tanto para conseguir la mejor configuración del algoritmo, como las pruebas referentes al enfrentamiento del algoritmo contra un jugador humano.
- Se extraerán las conclusiones adecuadas respecto de los resultados de las pruebas realizadas.

2.5 Conclusiones previas

Las conclusiones que se extraen de este proyecto son varias. La más importante es que el algoritmo desarrollado es una buena aproximación a un algoritmo con capacidad de jugar al Poker pero aún está lejos de ser capaz de vencer a un humano y alcanzar tal competitividad.

Para la creación de un mejor algoritmo se deberían de tener esencialmente dos factores presentes. El primero es la contemplación de mucha más información relativa al juego, pues factores como la agresividad de la mesa o la detección de faroles no se contemplan en el algoritmo.

Por otro lado se debería realizar infinidad de pruebas más para detectar la configuración de parámetros perfecta. Por ejemplo se deberían probar todas las configuraciones posibles entre los pesos e intercalarlas con diferentes valores de μ . También hacerlos para mayor cantidad de partidas por cada prueba. Esto podría llevar a un coste alto de tiempo por lo que si no se disponen de suficientes recursos se podría realizar una búsqueda heurística de las mejores configuraciones.

3 Desarrollo del TFM

3.1 Juego del Poker.

3.1.1 Definición

El póquer (también usaremos el término anglosajón 'poker') es un juego de cartas en el que pueden participar varios jugadores. En el juego se hacen apuestas sobre una puja común de modo que el ganador o ganadores de la partida se llevarán el monto final de la puja. Tradicionalmente el póquer se ha jugado con la baraja de tipo francesa (e inglesa, que es una adaptación de la francesa).

Durante una partida de poker (también lo llamaremos 'mano') se pueden ir eliminando jugadores en las sucesivas rondas de modo que el ganador de la partida será el que tenga la mejor combinación de cartas según una escala de valores que explicaremos en el punto 1.2.4. O bien el ganador puede ser el único superviviente que ha llegado hasta el final de la partida (o mano). Es posible que haya empates aunque éstos son poco frecuentes. Cuando dos o más jugadores empatan el monto de la puja es dividido y repartido en partes iguales por los jugadores ganadores.

Las apuestas en el poker pueden ser de dos tipos, con fichas sin valor monetario o con dinero real. Se recomienda sin ninguna duda a los que comiencen a practicar y aficionarse a este juego que empiecen jugando por fichas sin valor.

También existen dos modalidades de juego según la presencia física o no de los jugadores, es decir, se puede jugar presencialmente, con jugadores sentados en una mesa (normalmente) o de manera virtual, por ejemplo a través de una conexión en red con el software apropiado. [12][13]



(Imagen 7) momento típico durante una partida de Poker presencial donde se observan algunas cartas tapadas, otras cartas descubiertas y fichas por la mesa.

3.1.2 Historia

El origen del poker es un tema de debate en la actualidad y hay varias teorías sobre su creación e influencias. Algunas de ellas apuntan a que el póker proviene de la antigua china en el siglo X después de Cristo, siendo éste una variante del antiguo dominó chino. Otras teorías sugieren que el origen del poker es de un juego persa antiguo llamado “as nas” de modo que marineros persas exportasen este juego a la colonia francesa en norteamérica de la ciudad de Nueva Orleans.

El nombre de 'Poker' puede provenir del término francés 'poque', el cual también puede derivar del término alemán 'poche' que significa 'golpear', lo cual se relaciona a la práctica que sigue vigente hoy en día de pasar turno, que consiste en pegar dos golpes a la mesa haciendo entender a los demás jugadores que el jugador que ha golpeado no sube la apuesta y le pasa turno al siguiente jugador.

También se le relaciona con un juego del Renacimiento llamado “primero”, con otro juego francés llamado “blean” y con un juego inglés llamado “brag” que pudo aportar la práctica del “farol”, que consiste en aparentar tener mejores cartas de las que realmente tienes, aunque otras fuentes indican que ya se “faroleaba” en otros juegos anteriormente.

Cabe señalar que la popularidad del poker hoy en día está fuertemente relacionada con la introducción de este juego en el “viejo oeste” norteamericano, donde se convirtió en un juego muy popular que muchos colonos anglosajones en el nuevo mundo usaron para enriquecerse de manera rápida.

El juego llegó a declararse ilegal en el estado de Nevada, Estados Unidos, a principios del siglo XX, lo cual contrasta enormemente con la popularidad del Poker hoy en día en ese estado, principalmente en la ciudad de Las Vegas.

Actualmente el Poker es muy jugado en casinos de modo presencial y por aplicaciones online con jugadores de todo el planeta. [14]

3.1.3 Variantes

El póquer tiene muchas variantes hoy en día pero podemos agrupar todas sus variantes en tres familias principales en función a su protocolo de cartas y apuestas que son las siguientes:

Poker tapado:

Es uno de las modalidades de poker más tradicional. A cada jugador se le reparten cinco cartas y en ningún momento las cartas son enseñadas al resto de jugadores, es decir, las cartas están tapadas hasta el final de la partida. Durante la partida los jugadores tienen oportunidad de hacer un descarte, de modo que pueden cambiar algunas cartas para mejorar su estrategia. Hoy en día (año 2017 d.C) ya no es muy popular.

Poker descubierto:

Es la segundo tipo de poquer más antiguo, en esta modalidad algunas cartas son obligatoriamente descubiertas y otras tapadas. Hay varias rondas de apuestas y el orden para apostar no es siempre el mismo, sino que va en función de las cartas que tengan los jugadores. En esta modalidad no hay cartas comunitarias y se reparten 7 cartas (5 + 2 extra), de modo que al final de la partida cada jugador formará su mejor combinación de 5 cartas respecto las 7 que posee.

Poker con cartas comunitarias:

Es una variación del Poker descubierto, se reparten cartas tapadas a cada jugador y en el centro de la mesa se compartiran una serie de cartas comunes, de modo que todos los jugadores podrán usar las cartas del centro para formar combinaciones. Esta es la especialidad más difundida hoy en día y dentro de dicha variante están las subvariantes de Omaha y Texas Hold'em. En este proyecto trabajaremos con la modalidad Texas Hold'em.



(Imagen 8) Elección de la combinación de las 5 mejores cartas

3.1.4 Reglas del Poker Texas Hold'em

El Poker Texas Hold'em es una variación de Poker de la tipología de "cartas comunitarias". El juego se centra en apostar contra el resto de jugadores para ver quien tiene la mejor combinación de cartas .

Hay que diferenciar dos tipos de modalidad de juego dentro del Poker Texas Hold'em, aunque el objetivo y las reglas son las mismas, la finalidad de la partida puede variar diferentemente según si estemos jugando:

- Torneo
- Juego por valor (*cash game*)

El Poker Texas Hold'em se juega en una mesa con una cantidad de jugadores que puede variar desde 2 hasta 10.

El objetivo del juego es simple, ganar todas las fichas posibles que un jugador consiga. Hay dos maneras de ganar una mano o partida, o bien llegando hasta el final de la mano y poseyendo las mejores cartas, o bien consiguiendo ser el único que sobreviva al final de la mano.

La estructura de una mano o partida de Poker Texas Hold'em se puede dividir en 3 fases distintas:

- Configuración inicial de la mesa (*setup*)
- La ronda de apuestas (*betting rounds*)
- La muestra y comparación de las cartas (*showdown*)

Configuración inicial de la mesa:

Una vez los jugadores se han sentado alrededor de la mesa lo primero que necesitan es tener un conjunto de fichas con las que apostar. Se elige cual de los jugadores será el botón o *dealer* (de ahora en adelante usaremos el término anglosajón 'dealer'). Para elegir quien es el dealer se reparte una carta a cada jugador destapada, el jugador que tenga la carta más alta actuará de dealer. El dealer sirve para marcar el orden de los turnos para que cada jugador elija su opción. La posición del dealer irá rotando en cada partida hacia la izquierda, es decir, si B está sentado a la

izquierda de A y A es el dealer, en la próxima partida B será el dealer.

Una vez se ha establecido el dealer, pasamos a pagar las ciegas. El jugador que esté a la izquierda del dealer se encargará de pagar la ciega pequeña y el siguiente jugador a la izquierda pagará la ciega grande, la cual normalmente será el doble que la ciega pequeña.

El tamaño de las ciegas dictaminará las apuestas de la partida. Normalmente de inicio nadie entrará en la mesa a jugar con una cantidad de fichas que no supere 100 veces la ciega grande (no siempre se cumple, pero es lo recomendable). Una vez las ciegas ya se han pagado se pasa a la siguiente fase.

Ronda de apuestas:

La persona que reparte las cartas, comienza dándole la primera carta tapada al jugador que está a la izquierda del dealer y continúa repartiendo en el sentido de las agujas del reloj. Se para de repartir en el momento que todos los jugadores poseen 2 cartas.

Para entender bien el juego, hemos de tener presente que la partida finalizará cuando o bien todos los jugadores hayan abandonado a excepción de uno, o bien cuando llegemos a la última ronda de apuestas, sea completada y se comparen las cartas para ver que jugador tiene la mejor mano.

La primera ronda de apuestas es el *Preflop*.

El primer jugador en tomar una opción es el jugador que esté a la izquierda del que apostó la ciega grande. Hay 3 opciones (o acciones) para elegir:

- Abandonar la partida (*Fold*)
- Igualar la apuesta (*Call*)
- Subir la apuesta (*Raise*)

Al abandonar la partida el jugador pierde la oportunidad de ganar el monto final de fichas. Para igualar la apuesta el jugador debe pagar la ciega grande establecida. Para subir la apuesta el jugador debe pagar como mínimo el doble de la ciega grande. Por ejemplo, si la ciega pequeña son 5 fichas y la grande son 10 fichas, el jugador que hace *Raise* de inicio pagaría 20 fichas o más.

Una vez el jugador a elegido su opción, el turno pasa al jugador de su izquierda, de modo que para hacer Call siempre tendrá que pagar lo mismo que el jugador del turno anterior haya apostado, y para hacer Raise siempre habrá que pagar lo mismo que el jugador anterior haya apostado más una ciega grande.

Siguiendo el ejemplo anterior, si el primer jugador hubiera comenzado la partida apostando 20 fichas, el siguiente jugador que elige por primera vez, podrá retirarse de la partida (Fold), pagar 20 fichas para igualar (Call) o subir la apuesta (Raise) apostando desde 30 fichas (las 20 para igualar + 10 de la ciega grande) hasta todas las fichas que posea (a lo cual se le denomina *All-in*).

La ronda de apuestas terminará cuando se cumplan las dos condiciones siguientes:

- Todos los jugadores han tenido la oportunidad de actuar
- Todos los jugadores que no han abandonado han apostado la misma cantidad de fichas.

La segunda ronda de apuestas es el *Flop*.

Una vez la ronda del preflop ha terminado, se toma la baraja de cartas se descarta la primera carta sin mostrarla y se colocan destapadas en el centro de la mesa las 3 siguientes cartas (si jugamos Poker por medio de software, algunos detalles como descartar esa primera carta sin mostrarla son irrelevantes).

Una vez se han mostrado estas 3 cartas, comienza la ronda de apuestas.

Las reglas de las apuestas del flop son las mismas que las del preflop salvo dos excepciones:

- El primer jugador en hablar esta vez será el primero que está a la izquierda del dealer (el que pagó la ciega pequeña)
- Ese jugador no necesita pagar nada para seguir jugando, es decir, dado que la mesa ya está igualada en apuestas podrá hacer un Call gratuito, también llamado *Check* ('pasar' en castellano).

El resto de la dinámica del flop será la misma que la del preflop, el sentido de rotación de los turnos sigue siendo (siempre lo es) las agujas del reloj. Cuando la mesa esté igualada, se pasa a la siguiente fase.

La siguiente ronda de apuestas es el *Turn*.

Una vez el flop ha finalizado, se volverá a descartar la primera carta de la baraja, sin enseñarla a los jugadores, y después se tomará otra carta de la baraja y se dispondrá destapada en el centro de la mesa. Es decir, igual que en la fase de flop pero solamente destapando 1 carta en vez de 3. Se volverá a apostar y hacer la ronda de apuestas de la misma manera y orden que en el flop.

La siguiente ronda de apuestas es el *River*.

Asumiendo que más de un jugador sigue en la partida (esto se aplica a todas las rondas, de modo que si un jugador sube la apuesta, osea hace Raise y nadie le paga, la partida acaba y ese jugador se lleva el monto total apostado de la mesa), después de haber finalizado el Turn se vuelve a proceder de manera idéntica al Turn.

La única diferencia entre el River y el Turn es que el River es la última ronda de apuestas y el Turn la penúltima.

Muestra y comparación de las cartas (Showdown):

Una vez la ronda de apuestas del River ha finalizado entramos en la fase de comparar las cartas, para ello debemo tener presente los siguientes criterios:

- Quien tenga la mejor combinación de 5 cartas a partir de las 7 cartas posibles (las 2 de cada jugador más 5 de la mesa) es el ganador y se llevará el monto total de fichas de la mesa (puede haber empates, en tal caso se reparte el monto por partes iguales).
- El jugador que apostó en el River es el primero en mostrar sus cartas al resto de jugadores.
- Si nadie apostó en la ronda del River, el jugador más cerca a la izquierda del dealer es el primero en mostrar sus 2 cartas tapadas y por tanto revelar su combinación de cartas final.
- Si un jugador ya sabe que tiene una mano no ganadora tiene la opción de no mostrar sus cartas.

- Cada jugador no tiene la obligación de usar las 2 cartas tapadas para hacer la combinación de las 5 mejores cartas. Es decir, cada jugador puede usar ninguna, 1 o las 2 cartas tapadas que dispone y por tanto podrá usar 3, 4 o las 5 cartas de la mesa para hacer su combinación.
- Si dos jugadores empatan en su combinación de sus 5 mejores cartas, no se considera ni se analiza cuales son sus otras 2 cartas restantes.
- Si no hay jugador con una combinación de cartas, ganará el que tenga la carta de mayor valor. (la carta de mayor es el As, después el rey (K), la reina (Q), la J, el 10, el 9, el 8... hasta el 2.
- No hay ningún palo que valga más que otro. ♠♥♣♦

Una vez ha acabado la mano, el dealer pasa a ser el que está a la izquierda del anterior dealer (recordemos, se rota en sentido de las agujas del reloj) y se vuelven a repartir cartas si se desea seguir jugando. [15]

Combinación de mejores cartas, de mejor a peor combinación: [15]

Royal Flush	    	Escalera Real
Straight Flush	    	Escalera de Color
Four of a Kind	    	Póquer (4 cartas del mismo número)
Full House	    	Full (trío + pareja)
Flush	    	Color (5 cartas del mismo palo)
Straight	    	Escalera
Three of a Kind	    	Trío
Two Pair	    	Doble Pareja
One Pair	    	Pareja
High Card	    	Carta Alta

(Imagen 9) Ranking de cartas , ordenadas de más a menos valiosas

3.2 Relación del Poker con Teoría de Juegos

Expuestas anteriormente las categorías en las que podemos dividir los juegos en la teoría de juegos, vamos a especificar como éstas se aplican al Poker:

Según el número de jugadores:

En nuestro caso el mínimo será 2 y el máximo será establecido en 10 jugadores, a pesar de que según las reglas oficiales del Poker Texas Hold'em podrían jugar hasta 22 jugadores. Aunque en la práctica las mesas de Poker no suelen revasar los 10 jugadores.

Según el número de estrategias:

En el Poker las alternativas no son infinitas, a priori las alternativas son 3, y en una de ellas (la de subir la apuesta) hay que indicar numéricamente una cantidad que en todo caso estará acotada en un mínimo y máximo posible.

Según la relación entre los jugadores:

En el Poker Texas Hold'em las coaliciones están prohibidas, a pesar de que en la práctica algunos jugadores las hagan tanto explícita como implícitamente, pero el juego no las permite. Por tanto estamos tratando un juego sin coaliciones.

Según tipo de pago:

El Poker Texas Hold'em se catalogaría como juego de suma-cero. Pero con un matiz, cuando se juega con dinero real en casas de apuestas o casinos, éstos se llevan un pequeño porcentaje constante de lo apostado, por tanto la suma de ganancias y beneficios entre los jugadores en una partida no llega a ser nula. Sin embargo no siempre es así, por ejemplo cuando se apuestan fichas que no representan dinero ni valor real, en la práctica, no se pierde ningún porcentaje de fichas entre los jugadores, por tanto el juego sería suma-cero estrictamente. En nuestro caso no hay dinero real en juego, solo fichas ficticias o puntos, por tanto catalogamos el juego como "juego de suma-cero".

Según información disponible:

En el Poker Texas Hold'e no se conocen las cartas del resto de jugadores por tanto lo catalogamos como juego de información imperfecta.

Según la transferencia de información

Claramente el Poker Texas Hold'em se cataloga como juego competitivo, siendo además en nuestra modalidad tratada en este TFM como "estrictamente competitivo".

Según la aleatoriedad:

El Poker Texas Hold'em se catalogaría como "juego no determinista" y además "frente a la naturaleza" debido a la aleatoriedad de las cartas, tanto las que se reparten como las que van apareciendo a lo largo de una mano como también a las acciones de otros jugadores.

Según número de movimientos:

En el Poker Texas Hold'em se catalogaría de juegos de un paso, debido a que cada jugador tiene su turno para actuar, no hay simultaneidad.

Estrategias puras y mixtas:

En el Poker Texas Hold'em, establecer una estrategia eficiente y con una razonable probabilidad de éxito es difícil. Es decir, podríamos jugar con una estrategia simple como "siempre hacer ALL-In" o "siempre hacer CALL" o "siempre hacer 'faroles' ", sin embargo el fracaso a largo plazo de esta estrategia está garantizado. Por tanto, si hablamos en términos "simples", el Poker Texas Hold'em es un juego de estrategias mixtas.

Ahora bien, si establecemos un conjunto de estrategias refinadas y precisas, es probable que si las estrategias del conjunto están bien estudiadas y trabajadas, podría darse el caso que con una estrategia pura tengamos éxito. Sin embargo, en cuanto un oponente sea conocedor de nuestra estrategia (tras muchas manos de análisis y estudio) podría elegir (nuestro oponente) una estrategia que anule y por tanto venza a nuestra estrategia. Así pues, sin duda en el Poker Texas Hold'em se recomienda tener una estrategia mixta.

Un ejemplo claro de esto son los llamados "faroles" (farol, en la jerga del poker significa aparentar, por medio de tus apuestas, tener unas cartas mucho mejores de las realmente tienes. Un jugador que recurra muy asiduamente al recurso de los faroles probablemente acabe sin éxito. Y por otro lado, un jugador que nunca haga faroles es posible que tenga poco éxito. Muchos jugadores profesionales del Póker aconsejan usar el recurso del farol de vez en cuando.

Estrategias Dominantes:

Es difícil esclarecer esto en el Póker Texas Hold'em, pues como en otros juegos de cartas, a largo plazo se podría decir que no hay una estrategia dominante. Por ejemplo, si un jugador juega de manera conservadora acabará perdiendo contra otro jugador que sepa hacerle los llamados "faroles". Se denomina "Farol" en el argot o jerga del Póker al aparentar tener mejores cartas de las que en realidad se posee.

Estrategia Minimax y Maximin:

El Poker Texas Hold'em no es un juego de suma no-nula, es decir, es un juego de suma nula, por tanto no se contempla esta estrategia salvo en muy atípicas situaciones donde hay más de 2 jugadores y un tercero ya no puede hablar ni tomar acciones. En la mayoría de los casos solamente hay un jugador que obtiene ganancias, el resto de jugadores obtienen pérdidas. Por tanto la ganancia mínima no está garantizada dado que siempre hay un riesgo de pérdida en cada decisión que tomamos.

Sí podríamos hablar de estrategia Minimax, donde minimicemos la pérdida. Por ejemplo, si en una partida de Poker no tenemos muchas probabilidades de ganar la mano, es mejor no arriesgar más fichas y retirarnos (hacer FOLD), de ese modo minimizaremos la pérdida en esa mano.

Pareto optimo y equilibrio de nash

El concepto de Pareto óptimo no se puede aplicar al Poker Texas Hold'em. La política del bien común es contraria a la del Poker Texas Hold'em donde no importa el beneficio del grupo sino solo el del individuo.

Punto de silla:

En el Poker Texas Hold'em podríamos hablar de punto de silla en algunos casos muy concretos. Por ejemplo, en una mesa donde hay varios jugadores, podría darse el caso donde a 2 jugadores les convenga no subir más la apuesta y asegurarse un "empate" y acabar repartiéndose las ganancias de otros jugadores que ya no pueden hablar más en la mesa por haber hecho un all in de pocas fichas.

Teoria de la decision:

El Poker Texas Hold'em se deben tomar decisiones con incertidumbre pues no conocemos las probabilidades precisas de cada una de las opciones disponibles.

Por ejemplo, durante una partida de Poker nosotros podemos calcular la probabilidad de que la próxima carta que la mesa muestre nos sirva o no para formar una buena mano (por ejemplo una escalera o un full) sin embargo desconocemos las cartas que tienen nuestros oponentes, y por tanto desconocemos si ellos tienen opciones de vencer a nuestra posible buena mano (por ejemplo conseguimos sacar una escalera con la última carta, pero el rival consigue sacar escalera de color)

3.3 Estado del arte en el estudio científico del Poker

3.3.1 Introducción

Desde el comienzo de la computación se comenzó a buscar y especular con la programación de algoritmos que fueran capaces de ganar a humanos en juegos como el ajedrez, el conecta-4 o las damas entre otros.

En 1996, un acontecimiento revolucionó el mundo en la batalla intelectual entre las “máquinas y el ser humano” a partir del enfrentamiento que tuvieron el campeón mundial de ajedrez Garry Kasparov contra la computadora Deep Blue creada por IBM en un torneo de seis partidas de ajedrez.

En una primera disputa el jugador ruso de ajedrez venció a la computadora por 4 a 2, lo cual ya era bastante significativo. Pero además al año siguiente, en 1997 los desarrolladores de Deep Blue crearon una segunda versión llamada Deeper Blue que creó un precedente insólito en el mundo de la Inteligencia Artificial al vencer por 3 partidas a 2 al campeón del mundo eslavo. De ese Deeper Blue se convirtió en la primera computadora capaz de vencer al ajedrez al considerado mejor ajedrecista del planeta. [16]

Si bien es cierto que este acontecimiento era esperado, pues evidentemente tenía que ocurrir en algún momento ante el imparable avance del desarrollo de la inteligencia artificial que estaba y sigue teniendo. También es cierto que sirvió para, al menos popularmente, abrir la caja de pandora, de modo que en muchas otras disciplinas y otros juegos muchos desarrolladores se animaron a desarrollar algoritmos que fueran capaces de vencer a los seres humanos.

El Poker no se salva de ser otro juego afectado por este fenómeno, sino que incluso ha sido uno de los objetivos más populares del desarrollo de la inteligencia artificial seguramente por el aliciente que tiene de ser un juego de apuestas, en muchos casos con dinero real, de ese modo se aumentó el interés de muchos desarrolladores en intentar resolver el algoritmo o al menos en crear un algoritmo capaz de vencer a los humanos.

A diferencia del ajedrez, el Poker es la antítesis de los juegos de información perfecta pues el dato más relevante son las cartas del adversario, y este dato es justo el que se desconoce.

El poker es un juego con incertidumbre y por tanto encontrar una solución es mucho más complicado , complejo y costoso que lo que pueda ser para el ajedrez o las damas.

El póker es un juego donde no solo influyen las probabilidades, sino otros factores más relacionados con las emociones humanas. Para entendernos, es muy complicado programar un algoritmo que comprenda lo que es un farol y sepa cuando su contrincante lo está ejecutando. El comportamiento de un “bot de poker” siempre va a ser racional y el del humano no tiene por qué serlo (o sí).

Por otro lado, una máquina tiene sus ventajas frente al ser humano, como la de no sentir cansancio o fatiga y de ese modo no cometerá errores relacionados con ese aspecto como por ejemplo fallos de cálculo que un jugador humano podría cometer después de haber estado varias horas jugando a poker, o tampoco una máquina podrá estar en estado “tilt”.

El término Tilt en la jerga del poker hace referencia al estado que un humano toma de frustración o confusión cuando la estrategia con la que ha estado jugando, sea cual fuere, se torna no fructífera de modo que el jugador comience a jugar con una estrategia desmedida e irracional mucho más agresiva, por ejemplo haciendo “all in” reiteradamente (All in significa apostar todas las fichas que se poseen durante una mano).

A continuación vamos a hacer un breve repaso a diferentes proyectos de programas que juegan al poker automáticamente que han sido desarrollados principalmente en universidades anglosajonas con capacidad exitosa de vencer a humanos en el juego del Poker incluso algunos que afirman haber resuelto algunas variantes del Poker.

Antes de exponerlas debemos tener presente la diferencia entre Poker Texas Hold'em con límite y sin límite.

El Poker Texas Hold'em sin límite cuando un jugador realiza una apuesta(*raise*) el límite para apostar lo indica su cantidad de fichas, es decir que puede apostar todo lo que tiene en cualquier momento de la partida(a eso se le llama en jerga de poker (*all in*)).

Sin embargo en el Poker Texas Hold'em con límite las apuestas están limitadas a una cifra determinada, de ese modo las posibilidades para actuar en cada momento se ven sustancialmente reducidas y por tanto la resolución de este juego es más fácil de hallar.

3.3.2 Sistemas desarrollados

Polaris es un software con capacidad de jugar a Poker (llamaremos por simplicidad a este tipo de software como bot) en su modalidad de Texas Hold'em con límite desarrollado en la universidad de Alberta, Canadá, desarrollado en 2007 que a la vez contiene diferentes modalidades capaces de jugar al poker con diferentes estrategias. Una de ellas es "Hyperborean08" la cual participó y ganó una competición de bots organizada en 2008 por la Asociación por el desarrollo de la inteligencia artificial (Association for the advancement of Artificial Intelligence).

Polaris posee un conjunto de diferentes estrategias y durante las partidas de poker va cambiando la estrategia. El coste computacional de Polaris no es excesivo de modo que puede ser ejecutado en una computador comercial o convencional.

Una característica esencial que debemos tener presente es que Polaris está diseñada para competir en modo "*heads-up*", es decir, contra solo un adversario, lo cual es muy importante subrayar debido a que esta modalidad no es la habitual y tradicional en el poker aunque también sea practicada.

Respecto a competir contra jugadores profesionales humanos, Polaris fue puesta a prueba al menos en dos ocasiones. La ocurrida en el año 2007 consistió en cuatro torneos (entendiendo por torneo un conjunto de manos seguidas) de 500 manos cada una donde a los jugadores (humanos y bots) se les dieron exactamente las mismas cartas a lo largo del torneo pero en diferentes momentos. De ese modo se eliminaba el factor suerte (es decir, si durante un torneo el humano recibía en 10 ocasiones dos Ases como cartas, al bot también se les suministrarían esos 10 Ases. El resultado fue que el jugador humano ganó 2 partidas a 1, y empataron otra. [17]

Sistema Cepheus

El proyecto Cepheus es un algoritmo desarrollado que juega a poker que según sus creadores, esencialmente ha resuelto el juego de poker en su variedad de Poker Texas Hold'em Heads-up con límite.

A partir de ahí podría considerarse como uno de los gran hitos de la computación en teoría de juegos aplicada al poker. Esto es así debido a que se proclamaría como el primer algoritmo que resuelve un juego de información imperfecta.

Cepheus fue desarrollado durante una década de investigación por grupos de investigación de la universidad de Alberta, en Canadá, al igual que su antecesor Polaris. Además participó en el proyecto el finlandés Oskari Tammelin con la implementación del algoritmo CFR+ que más tarde explicaremos.

En enero de 2015 fue publicado en la revista "*Science*" el artículo "*Heads-up limit hold'em poker is solved*" donde se explica como Cepheus resuelve el juego.

La estrategia de Cepheus se acerca al equilibrio de Nash de modo que una contraestrategia solo podría ganar 0,000986 ciegas grandes por juego. Es por ello que los desarrolladores nos anuncian que "esencialmente" Cepheus resuelve el juego dado que para estar completamente resuelto habría que descender el dato 0,000986 ciegas por juego a simplemente 0.

Aún así, los desarrolladores afirman que Cepheus es tan cercano a la perfección que si un jugador humano de poker estuviese jugando toda su vida (60 millones de manos) contra Cepheus no podría notar sustancialmente la diferencia con la estrategia perfecta.

El algoritmo se desarrolló sin experiencia humana, sencillamente conocía las reglas del juego y se entrenó jugando contra si mismo una y otra vez hasta más de un billón de partidas en cada segundo durante 2 meses, usando en el proyecto más de 4 mil CPUs. Para hacernos una idea de la dimensión del número de partidas, en toda la historia de la humanidad no se han jugado tantas partidas como las que usó Cepheus para entrenarse.

CFR+

Cepheus utiliza el algoritmo CFR+ desarrollado a partir de CFR. Como ya adelantamos, este algoritmo aprende a jugar simplemente entrenándose contra si mismo. El algoritmo empieza jugando manejando una serie de probabilidades random. A partir de los resultados obtenidos el algoritmo va modificando dichas probabilidades para mejorar su estrategia de modo que poco a poco se va acercando a la estrategia óptima.

La manera en que esto funciona es valorando con una cantidad de *regret* o *feedback* cada acción que ha tomado tras un momento de decisión. Es decir, el algoritmo analiza como de bien le hubiese ido si desde que comenzó a jugar hubiese tomado esa decisión concreta en vez de haber estado tomando la otra estrategia que el algoritmo tenía como apropiada y ha usado hasta ese momento.

Los desarrolladores de Cepheus han habilitado una web para que se pueda jugar online contra el algoritmo en la siguiente página web:

<http://poker-play.srv.ualberta.ca/> [18] [19] [20]

Claudico

Claudico es otro jugador bot de Poker diseñado y desarrollado por la Universidad Carnegie Mellon en Pittsburgh, Estados Unidos. Su nombre proviene de la jerga de poker "*limp*" que significa pagar la apuesta de un contrincante.

A diferencia de Polaris y Cepheus. Claudico está diseñado para jugar a la variante de poker Texas Hold'em Heads-up sin límite, que como ya dijimos es mucho más difícil de desarrollar de manera exitosa.

Una de las peculiaridades de Claudico es que usa una estrategia totalmente propia, es decir, no se basa en unas directrices o unas estrategias fijas que un jugador de poker pueda implementar en el algoritmo (como se han diseñado otros algoritmos) sino que el mismo bot desde el principio crea y moldea sus propias estrategias.

El coste computacional de este programa es alto, lo suficiente como para necesitar de 16 terabytes de memoria RAM para poder ejecutarse.

Al igual que Polaris, y Cepheus, Claudico está diseñado para jugar en la modalidad de poker de "heads-up", es decir, una mesa de uno contra uno.

En el año 2015 Claudico se puso a prueba con jugadores profesionales de Poker del circuito internacional. Entre ellos figuraba Doug Polk, que entonces era el número uno del ranking mundial en la modalidad de Poker de Heads-up.

El torneo se disputó con cuatro jugadores profesionales más el jugador bot Claudico. Se jugaron cerca de 20 mil manos, con las mismas manos a lo largo de todo el torneo. Finalmente Claudico quedó en cuarta posición, por tanto 3 jugadores profesionales le superaron y uno quedó en peor posición que el bot.

Doug Polk comentó sobre la forma de jugar de Claudico que hizo algunas jugadas muy extrañas que un humano nunca haría, como arriesgar 19 mil fichas para ganar 700.

Miembros de los desarrolladores de Claudico afirma que el algoritmo empleado que maneja información incompleta puede tener usos en la ciencia médica y en la seguridad.[21] [22]

Libratus:

Libratus es una versión mejorada de Claudico, por la Universidad Carnegie Mellon en Pittsburgh, Estados Unidos. Pero a diferencia del anterior, Libratus se contruyó con más de 15 millones de horas de computación, sin embargo Claudico se contruyó con 2 o 3 millones.

Libratus usa un variante del método CFR+, muy popular en la implementación de jugadores bots, creada en 2014 por Oskari Tammelin. A esta variante se le denominó “*action mapping*”.

Al igual que su hermano pequeño, Claudico, está diseñado para jugar la modalidad de Poker de “*Heads up*” (solo 2 jugadores en la mesa).

En enero de 2017 Libratus fue puesto a prueba contra 4 jugadores profesionales de Poker en la modalidad de *Heads up*. A diferencia de la competición anterior donde participó Claudico, en esta ocasión se incrementaron el número de manos y los días de juego del torneo, pasando de 13 días a 20.

Los resultados fueron muy sorprendentes y chocantes pues el bot Libratus ganó a todos los demás jugadores humanos por una diferencia enorme y aplastante. Había 2 millones de fichas en juego, a disputar entre los 4 jugadores profesionales y el bot. Libratus se hizo con más de la mitad de las fichas doblando en número de fichas al segundo clasificado.

Uno de los participantes del torneo que compitió contra Libratus afirmó que tenía la sensación de que el bot conocía las cartas que el tenía.

Al igual que con su versión anterior Claudico, los desarrolladores norteamericanos afirmaron que el uso de este algoritmo puede ser usado en otras áreas no relacionadas con el poker donde se tratan con problemas de información incompleta. [22] [23]

3.4 Propuesta de algoritmo

3.4.1 Aproximación inicial

Inicialmente habrá que crear una plataforma que permite jugar al *Poker Texas Hold'em*, y que acepte tanto jugadores humanos como jugadores máquina, bots.

Además de la plataforma, se desea desarrollar un agente artificial, es decir, un jugador de Poker cuyas decisiones sean tomadas según los cálculos y resultados de un algoritmo con capacidad de calcular la mejor decisión en cada turno de un agente.

Para dicho objetivo el algoritmo debe cuantificar numéricamente, con valores reales, cual es la información general de la mesa en un momento preciso. En la siguiente sección, detallaremos la información relevante a cuantificar.

El agente debe tener capacidad de aprendizaje de modo que su estrategia de juego sea adaptativa a las circunstancias de cada momento y se base en la búsqueda de la mínima pérdida.

3.4.2 Información a considerar

La información a considerar por parte del algoritmo para poder jugar y tomar una decisión serán datos típicos de una partida de Poker. Intentaremos emular el comportamiento humano en una partida de *Poker Texas Hold'em Sin límite*, el algoritmo se centra en los datos que un humano más importancia otorga en el momento de tomar una decisión durante una partida de Poker.

Esos datos de partida en los que el algoritmo tratará de basarse para tomar una decisión son los siguientes:

- Lo buenas o malas que sean las cartas que el jugador bot posee
- La posición en la mesa del jugador bot
- la cantidad de fichas que hay que apostar para optar a ganar el bote de la partida
- la cantidad de fichas que le quedan al bot para jugar en relación a las que hay en toda la mesa

3.4.3 Estrategias del juego

Para elaborar la estrategia del algoritmo durante las partidas de Poker, se tendrá en cuenta cuál es el comportamiento humano en una partida de Poker, y como suele modificar un humano su estrategia durante las partidas de Poker.

Se puede afirmar que a grandes rasgos, los jugadores humanos competentes en Poker suelen tener un rango de estrategia que se mueve entre dos polos. Por un lado tendríamos un estilo de juego conservador que definiríamos como :

- Estrategia conservadora: dícese de la estrategia que se centra en no perder fichas de modo que su tendencia a jugar es siempre sin riesgo, o con riesgo mínimo. Esta estrategia se caracteriza por no perder muchas fichas pero suele estar lejos de conseguir grandes victoria en las partidas y por tanto lejos de ganar grandes botes. (bote, refiérese a un conjunto de fichas que se otorgan como premio al vencedor de una partida o mano)

- Estrategia arriesgada: dícese de la estrategia que se centra en ganar las partidas o manos de Poker y para ello apuesta, y por tanto arriesga grandes sumas de fichas. Esta estrategia se caracteriza por ganar grandes botes en algunas partidas pero también por perder muchas fichas en otras partidas.

Nuestro algoritmo se moverá entre el rango establecido de esas 2 estrategias. Para ello crearemos 2 parámetros, que definirán como de arriesgado se está jugando o como de conservador se está jugando.

3.4.4 Propuestas de aprendizaje

El algoritmo debe ser capaz de aprender durante las partidas, por tanto su estrategia irá variando y lo hará conforme a los resultados que va obteniendo.

Para conseguir este objetivo establecemos 2 puntos fundamentales:

- el algoritmo será configurable, es decir, el algoritmo tendrá la capacidad de tener una serie de parámetros con valor variable que serán ajustados según la necesidad y situación de cada momento.
- el algoritmo revisará automáticamente los resultados cada X partidas, de modo que si los resultados están siendo buenos, el algoritmo mantendrá la misma estrategia, pero si los resultados son malos, el algoritmo modificará la estrategia.

Para modificar la estrategia se ajustarán los parámetros anteriormente comentados, que provocan que la estrategia del algoritmo sea o bien más conservadora, o bien más arriesgada.

4 Diseño del Algoritmo

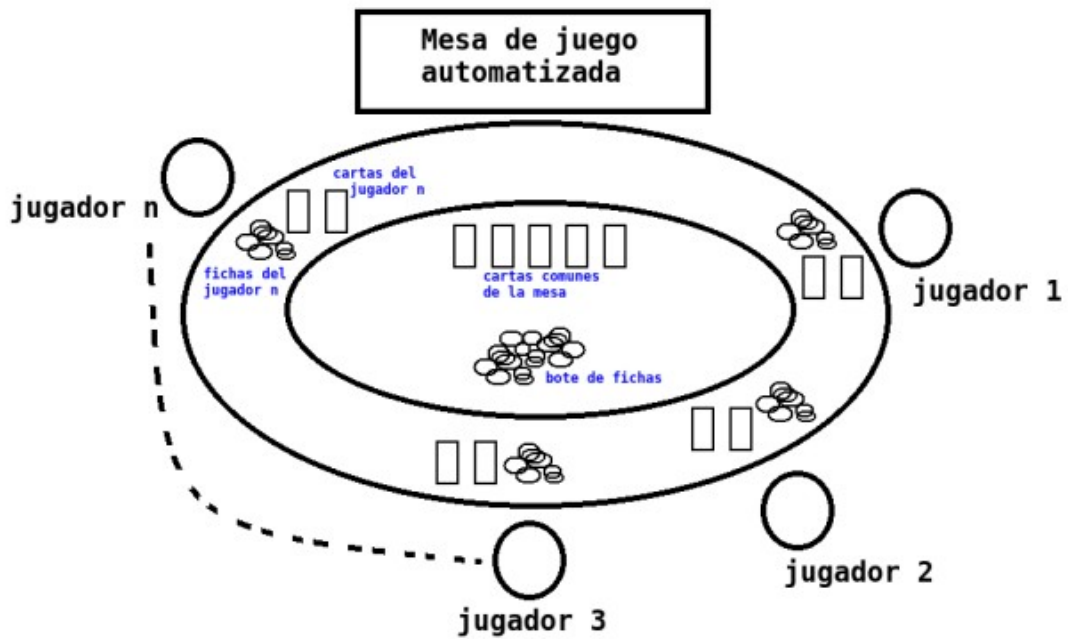
4.1 Introducción general

Para el desarrollo de este TFM de necesita elaborar una plataforma que permita el juego simultáneo de jugadores máquinas (bots) y jugadores humanos en una partida de *Poker Texas Hold'em Sin límites*.

La plataforma debe cumplir con todas las reglas del Poker en dicha variante, lo cual no solo incluye las reglas en los turnos, el reparto de cartas, y el sistema de apuestas, sino además incluye el reparto de ganancias entre los ganadores, o el ganador, según sea el caso, la actualización de fichas de todos los jugadores al terminar una partida y el poder jugar en dos modalidades:

- modo simple: los jugadores de la mesa siempre pueden jugar de modo que si se quedan fichas puedan recargarse de fichas y continuar jugandol (sistema que será usado para el entrenamiento de los jugadores bots)
- modo torneo: comienza la partida con un número fijo de jugadores, Si un jugador queda eliminado no puedo volver a jugar hasta que finaliza el torneo. El torneo se termina cuando solamente queda un jugador en la mesa con todas las fichas.

Las mesas de juego serán de 2 a 10 jugadores como mínimo y máximo respectivamente. A continuación se dibuja un esquema conceptual de como será la resultante mesa que proporcione la plataforma que se debe crear.

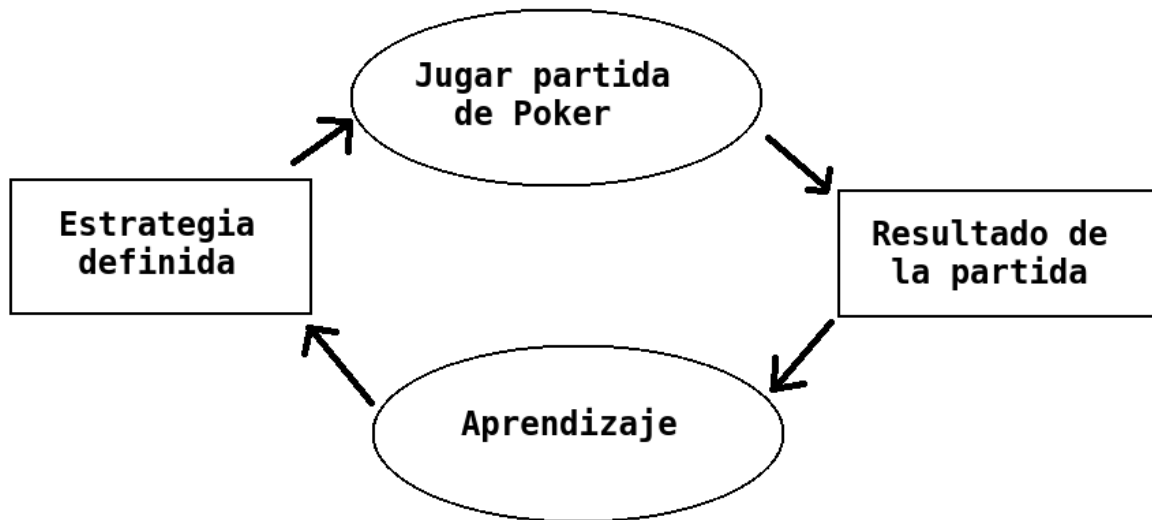


(imagen 9) Mesa de juego proporcionada por la plataforma

Como vemos en la imagen 9, los jugadores se sentarán de manera ordenada, uno tras otro, y los turnos de juego siempre respetarán dicho orden.

Para el desarrollo del algoritmo vamos a modelar de modo automático como debe ser el comportamiento de juego de uno de los jugadores de la mesa. El algoritmo resultante se podrá implementar en varios jugadores si se desea, de modo que en una mesa pueda haber desde 0 a N jugadores bots, y lo mismo con los jugadores humanos.

A continuación se representa en la imagen 10, a nivel general, el funcionamiento del algoritmo implementado en un jugador máquina, es decir en un bot.



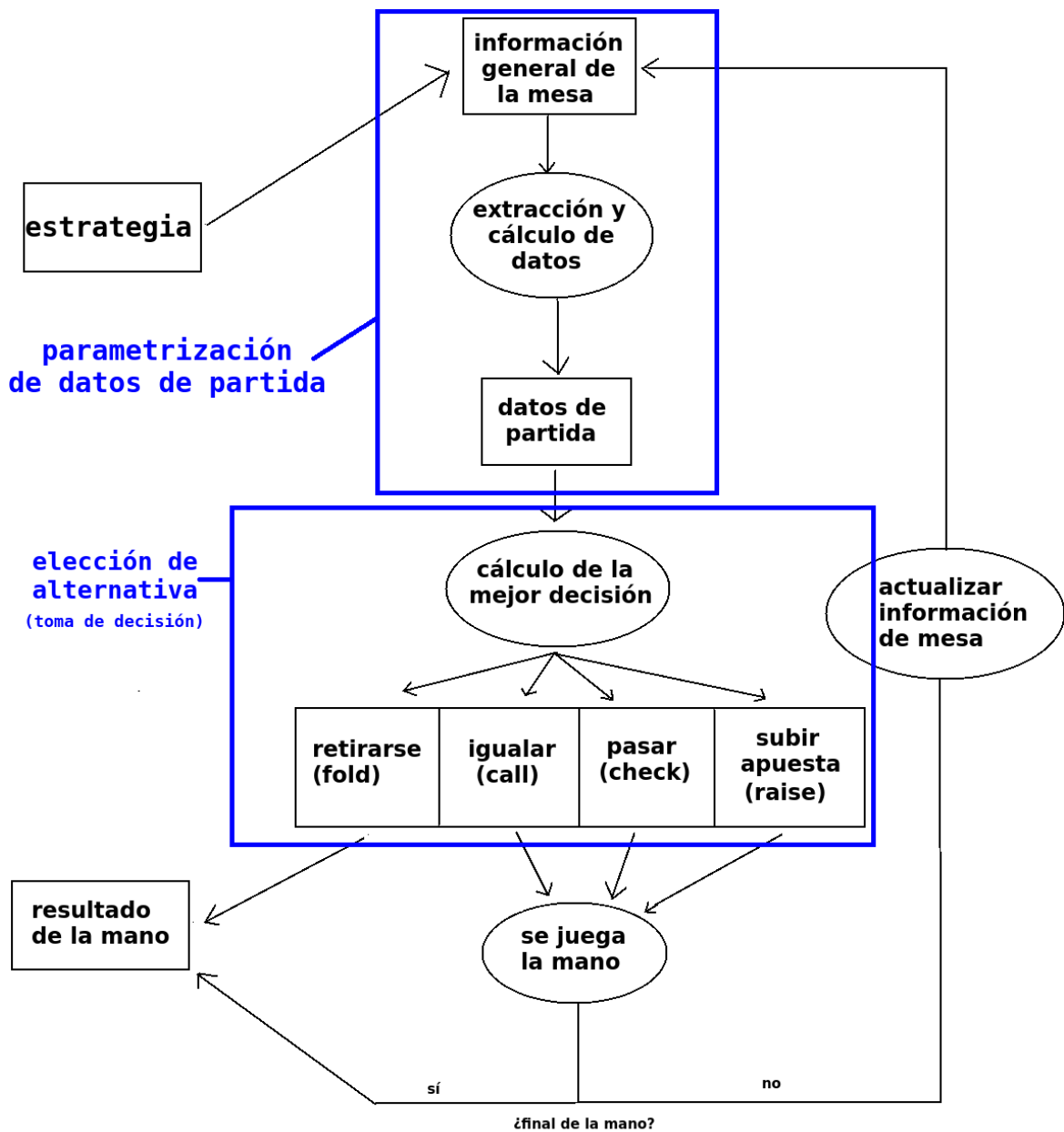
(Imagen 10) Funcionamiento general de un jugador máquina (bot)

De este modo, se diferencian dos procesos distintos que desgranaremos en los siguientes apartados, por un lado tenemos el proceso de “Jugar una partida de Poker”, donde el algoritmo tendrá que parametrizar valores, y elegir en cada momento sobre las alternativas posibles.

Y por otra parte tendremos el proceso de “Aprendizaje”, donde el algoritmo en caso de obtener malos resultados, ajustará su estrategia de juego para conseguir mejorar dichos resultados.

4.2 Jugar partida

Para que un jugador máquina juegue una partida de Poker, el algoritmo desarrollado debe marcar en todo momento los pasos a seguir y tener una capacidad de decisión inteligente en el momento de tomar una decisión. Para desgranar el algoritmo, se debe desarrollar más el esquema del proceso “jugar partida” de la Imagen 9, y así se podrá comprender más su comportamiento:



(Imagen 11) proceso del algoritmo durante el juego de una partida

4.2.1 Parametrización de los datos de partida

La primera parte del proceso de “Jugar Partida” es la parametrización de los datos, es decir, se toman todos los datos objetivos y descriptivos de la mesa y a partir de ellos se calcula la utilidad de las posibles alternativas a elegir.

Para ello se debe analizar la información que interesa recopilar, la cual llamaremos “Datos de partida”:

Los datos de partida se extraen de la situación del juego en un momento preciso. Es decir, mientras varios bots o humanos están jugando una mano el programa genera toda una información pública respecto a la mesa de juego (como el bote, las cartas comunes, las fichas de los jugadores...), así como también una información privada para cada jugador (que son sus 2 cartas tapadas). A partir de toda esa información el algoritmo extraerá ciertos datos concretos que le interesa para usarlos en el cálculo de otros datos más específicos, que son los que se llaman “datos de partida”.

Se debe tener presente que esta información es objetiva, es decir, es una descripción numérica de la realidad del juego en ese momento.

Estos datos de partida que se calculan a partir de la información general de la mesa servirán para calcular la utilidad del jugador en un momento concreto durante una mano.

Para ello, se va a comentar primero la información que se necesita extraer de la mesa para hacer posteriores cálculos.

La información de la mesa que necesitará el algoritmo para calcular los datos de partida es la siguiente:

- Información sobre las cartas

Hay 2 tipos de cartas, las privadas que son personales y solo el jugador puede saber cuales son, y las cartas públicas que son comunes a todos los jugadores y éstos pueden usarlas

para realizar combinaciones con sus cartas.

- Información sobre la posición de los jugadores en la mesa

El dealer es el que tiene la mejor posición por ser el que “habla” en último lugar, dado que éste cuando tenga que tomar su decisión ya conocerá la decisión del resto de jugadores. El que tiene la peor posición es el primero en “hablar”.

- Información sobre las fichas de la mesa

Se debe diferenciar entre las fichas que un jugador tiene en posesión (a eso se le llama el *stack* del jugador) y por otro lado las fichas que están en la mesa que son las que los jugadores han ido apostando, que llamaremos el bote de la mesa, éstas últimas fichas, el bote, constituye el premio final que se llevará el ganador o los ganadores de esa mano.

- Información sobre los jugadores de la mesa

Refiere a los jugadores que están jugando en la mesa, o los que durante una mano estén aún con opciones de ganar la mano. En ciertos momentos también interesará diferenciar entre el número de jugadores en un determinado momento que tienen opción a ganar la mano (osea a llevarse el bote) y los jugadores que tienen opción a “hablar”, osea a realizar más apuestas en una mano. Puede ocurrir que un jugador tenga aún opciones de ganar la mano y llevarse el bote pero no tenga opción a apostar más porque apostó todas las fichas que tenía.

A partir de esta información que nos proporciona la mesa, cada jugador bot calculará y hallará una serie de datos precisos , los “datos de partida”. La suma ponderada de estos datos de partida determinará el potencial que se tiene para ganar una mano, es decir, su utilidad.

Como se adelantó anteriormente, la mesa nos proporciona una información general de la partida, y a partir de esa información nuestro algoritmo calculará los datos de partida, los cuales servirán para calcular el potencial que tiene el bot en ese instante de la mano. Es decir, para calcular la utilidad.

Cada dato de partida tendrá una valoración entre 0 y 1.

Los 4 datos de partida que tenemos que calcular son los siguientes:

- Valor de nuestras cartas (V)

- Posición de la mesa (P)

- Bote (B)

- Stack de un jugador (S)

A continuación explicaremos cómo calcular estos 4 datos:

Valor de nuestras cartas, V.

Este valor depende de en qué fase de la mano estemos.

En la fase de preflop se poseen solo 2 cartas, así pues V se calculará automáticamente de una tabla predeterminada de valores donde AA son las cartas de valor máximo y 7-2 son las cartas de valor mínimo. El rango de V está entre 0 y 1.. [24]

valor de V en la fase de PREFLOP:

Posición	Cartas	Valoración			
1	AA	1,00	11	A K	0,9345
2	K K	0,9881	12	A 10s	0,9286
3	Q Q	0,9821	13	Q Js	0,9226
4	A Ks	0,9762	14	K 10s	0,9167
5	J J	0,9702	15	Q 10s	0,9107
6	A Qs	0,9643	16	J 10s	0,9048
7	K Qs	0,9583	17	9 9	0,8988
8	A Js	0,9524	18	A Q	0,8929
9	K Js	0,9464	19	A 9s	0,8869
10	10 10	0,9405	20	K Q	0,8810

Continuación de valor de V en la fase de PREFLOP

21	8 8	0,8750	61	Q 7s	0,6369
22	K 9s	0,8690	62	8 6s	0,6310
23	10 9s	0,8631	63	6 5s	0,6250
24	A 8s	0,8571	64	J 7s	0,6190
25	Q 9s	0,8512	65	5 4s	0,6131
26	J 9s	0,8452	66	Q 6s	0,6071
27	A J	0,8393	67	7 5s	0,6012
28	A 5s	0,8333	68	9 6s	0,5952
29	7 7	0,8274	69	Q 5s	0,5893
30	A 7s	0,8214	70	6 4s	0,5833
31	K J	0,8155	71	Q 4s	0,5774
32	A 4s	0,8095	72	Q 3s	0,5714
33	A 3s	0,8036	73	10 9	0,5655
34	A 6s	0,7976	74	10 6s	0,5595
35	Q J	0,7917	75	Q2s	0,5536
36	6 6	0,7857	76	A 9	0,5476
37	K 8s	0,7798	77	5 3s	0,5417
38	10 8s	0,7738	78	8 5s	0,5357
39	A 2s	0,7679	79	J 6s	0,5298
40	9 8s	0,7619	80	J 9	0,5238
41	J 8s	0,7560	81	K 9	0,5179
42	A 10	0,7500	82	J 5s	0,5119
43	Q 8s	0,7440	83	Q 9	0,5060
44	K 7s	0,7381	84	4 3s	0,5000
45	K 10	0,7321	85	7 4s	0,4940
46	5 5	0,7262	86	J 4s	0,4881
47	J 10s	0,7202	87	J 3s	0,4821
48	8 7s	0,7143	88	9 5s	0,4762
49	Q 10	0,7083	89	J 2s	0,4702
50	4 4	0,7024	90	6 3s	0,4643
51	3 3	0,6964	91	A 8	0,4583
52	2 2	0,6905	92	5 2s	0,4524
53	K 6s	0,6845	93	10 5s	0,4464
54	9 7s	0,6786	94	8 4s	0,4405
55	K 5s	0,6726	95	10 4s	0,4345
56	7 6s	0,6667	96	10 3s	0,4286
57	10 7s	0,6607	97	4 2s	0,4226
58	K 4s	0,6548	98	10 2s	0,4167
59	K 3s	0,6488	99	9 8	0,4107
60	K 2s	0,6429	100	10 8	0,4048

Continuación de valor de V en la fase de PREFLOP

101	A 5	0,3988	136	6 4	0,1905
102	A 7	0,3929	137	Q 6	0,1845
103	7 3s	0,3869	138	5 3	0,1786
104	A 4	0,3810	139	8 5	0,1726
105	3 2s	0,3750	140	10 6	0,1667
106	9 4s	0,3690	141	Q 5	0,1607
107	9 3s	0,3631	142	4 3	0,1548
108	J 8	0,3571	143	Q 4	0,1488
109	A 3	0,3512	144	Q 3	0,1429
110	6 2s	0,3452	145	7 4	0,1369
111	9 2s	0,3393	146	Q 2	0,1310
112	K 8	0,3333	147	J 6	0,1250
113	A 6	0,3274	148	6 3	0,1190
114	8 7	0,3214	149	J 5	0,1131
115	Q 8	0,3155	150	9 5	0,1071
116	8 3s	0,3095	151	5 2	0,1012
117	A 2	0,3036	152	J 4	0,0952
118	8 2s	0,2976	153	J 3	0,0893
119	9 7	0,2917	154	4 2	0,0833
120	7 2s	0,2857	155	J 2	0,0774
121	7 6	0,2798	156	8 4	0,0714
122	K 7	0,2738	157	10 5	0,0655
123	6 5	0,2679	158	10 4	0,0595
124	10 7	0,2619	159	3 2s	0,0536
125	K 6	0,2560	160	10 3	0,0476
126	8 6	0,2500	161	7 3	0,0417
127	5 4	0,2440	162	10 2	0,0357
128	K 5	0,2381	163	6 2	0,0298
129	J 7	0,2321	164	9 4	0,0238
130	7 5	0,2262	165	9 3	0,0179
131	Q 7	0,2202	166	9 2	0,0119
132	K 4	0,2143	167	8 3	0,0060
133	K 3	0,2083	168	8 2	0,0000
134	9 6	0,2024	169	7 2	0,0000
135	K 2	0,1964			

(Imagen 12) Valor de las manos en la fase de Preflop

En la imagen 12, la *s* significa *suited*, es decir, que las dos cartas son del mismo palo. Las parejas de cartas suited tienen siempre mejor valoración que las que no. Esto es debido a que las *suited* aumentan sus posibilidades de hacer color o escalera de color.

En los casos de estar en la fase de Flop, Turn y River, tendremos 5 , 6 y 7 cartas respectivamente por cada jugador (2 cartas propias, más las de la mesa). En esos casos lo que valoraremos es la jugada que llevemos en la mesa.

Valoración de V según la combinación de cartas
que posea un jugador bot en las fases
de Flop, Turn o River

Combinación de cartas	Valoración
Escalera de color	[1, 0.9]
Poker	[0.9 , 0.8]
Full	[0.8 , 0.7]
Color	[0.7 , 0.6]
Escalera	[0.6 , 0.5]
Trio	[0.5 , 0.4]
Doble Pareja	[0.4 , 0.3]
Pareja	[0.3 , 0.2]
Carta Alta	[0.2 , 0.1]

El valor exacto de la combinación de cartas dentro de ese rango dependerá de las cartas que tenga. Por ejemplo:

Un Poker de Ases tendrá una valoración mayor que un Poker de treses.

$V(A, A, A, A) > V(3, 3, 3, 3)$; pero ambos están en el rango de Valoración de Poker [0.9 , 0.8]

Un Full de Dieces y Jotas (10, 10, 10, J, J) estará por encima que un full de Cincos y Ases (5, 5, 5, A, A)

$(10, 10, 10, J, J) > V(5, 5, 5, A, A)$

Posición en la mesa, P

Es un valor que varía según la posición a la hora de tomar una decisión durante una mano. La mejor posición será la de hablar el último y la peor la de hablar el primero. Este valor comprende entre 0 y 1 y se calcula en función al número de jugadores con capacidad de decidir durante una mano. (es posible que haya un jugador que tenga posibilidad de ganar el bote de una mano pero no tenga más opción de “hablar” debido a que ya apostó todo lo que tenía (all-in)).

Cálculo de P:

$$P = \text{Pos} / \text{NumJugHab}$$

NumJugHab = Número de jugadores que pueden hablar en ese turno

Pos = posición en el orden de hablar. Por ejemplo si hay 3 jugadores, el primero en hablar tendrá posición 1 y el última en hablar el 3.

De tal modo, cuando haya 2 jugadores en mesa que puedan tomar decisiones la P tendrá los siguientes valores:

Posición	NumJugHab	P
1	2	0,5
2	2	1

Para 3 jugadores:

Posición	NumJugHab	P
1	3	0,333
2	3	0,666
3	3	1

Y así sucesivamente.

Bote de la mesa, B

Es la proporción entre lo que un jugador tiene que apostar para poder seguir jugando y el bote que hay en la mesa en juego. Cuanto mayor sea el bote de la mesa respecto a lo que haya que seguir apostando, más merecerá la pena ir a jugar, por tanto mayor será B. Sin embargo si la cantidad de fichas que hay que apostar para seguir jugando es alta y el bote de la mesa no es tan alto, menos merecerá la pena arriesgarse a pagar y seguir jugando. El valor de B estará entre 0 y 1.

El cálculo de B es el siguiente:

$$\mathbf{B = 1 - (ApuesM / Bote)}$$

ApuesM = Apuesta mínima que hay que pagar para seguir jugando

Bote = Número de fichas en juego que hay en la mesa.

ApuestaM nunca podrá ser mayor que Bote:

Siempre se cumple que $\text{ApuestaM} \geq \text{Bote}$

Stack disponible en proporción, S

El stack es la cantidad de fichas que nosotros tenemos disponibles para apostar. S será la relación entre nuestro stack y la cantidad de fichas que hay en la mesa, es decir, la suma total de fichas de jugadores. Cuanto mayor sea el porcentaje de fichas de la mesa que un jugador tiene, más probabilidades hay de que este jugador juegue. En el mundo del poker hay consenso en afirmar que suele tener ventaja el jugador que tiene más fichas debido entre otras razones, a que en cada mano arriesga menos fichas que el resto en proporción a las que tiene.

Para ponderar este valor entre 0 y 1.

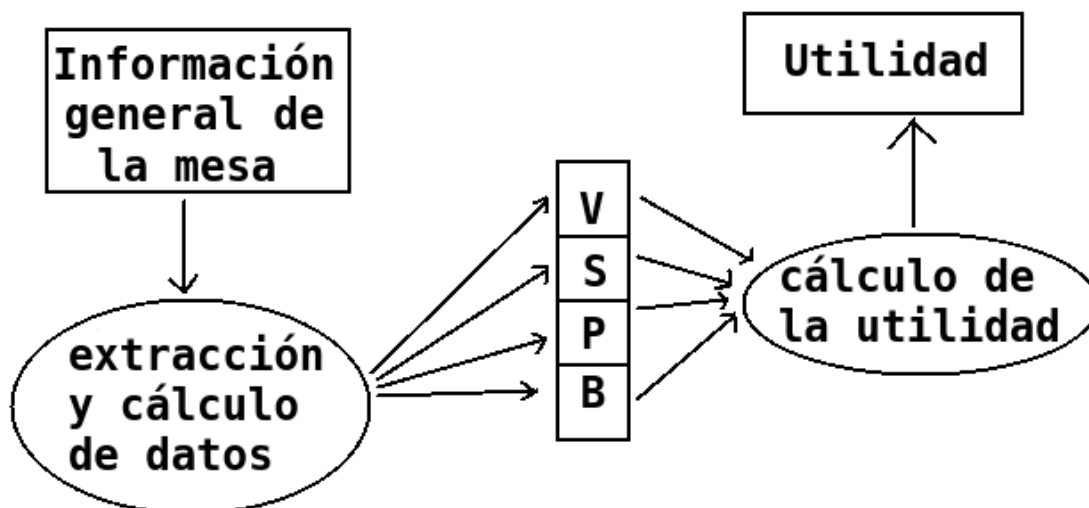
$$S_i = \text{Stack}_i / \sum \text{Stack}_j$$

j = cada uno de los jugadores de la mesa

S_i = valor S del jugador i

Stack_i = Cantidad de fichas del jugador i

Importante: el cálculo de la utilidad se realiza en función de estos 4 datos de partida



(Imagen 13) Proceso de la parametrización de los datos de partida

4.2.2 Elección de alternativa. Toma de decisión

Durante una mano de Póker existen varios instantes en los que un jugador tiene que decirle a la mesa (osea al resto de jugadores o al sistema que controla la partida) que decisión de las posibles alternativas va a tomar.

En Poker Texas Hold'em hay 4 posibles alternativas:

- Subir la apuesta (*Raise*)
- Igualar la apuesta (*Call*)
- Pasar (*Check*) (se podría entender como un Call con un coste de 0 fichas)
- Retirarse de la mesa (*Fold*)

Pero nunca un jugador se enfrenta a la decisión de elegir una de esas alternativas. La decisión puede ser entre 2 alternativas (*Raise* y *Check*) o entre 3 alternativas (*Raise*, *Call* o *Fold*).

Antes de describir este último punto más ampliamente, tengamos presente que los jugadores bots considerarán solo las alternativas racionales. Por ejemplo, si llega el turno de un jugador, y la cantidad de fichas que se le exige para seguir es 0, es irracional hacer *Fold* y retirarse de la partida. Por tanto descartaremos esa alternativa.

Así pues, durante el juego de una mano de poker a un jugador se le pueden presentar las 2 siguientes situaciones:

Caso A)

La apuesta mínima para continuar en la mesa es 0 por tanto

Alternativa 1: hacer **Check**, el jugador continúa en la partida sin apostar ficha alguna y el turno pasa al siguiente jugador.

Alternativa 2: hacer **Raise**, el jugador sube la apuesta en X y pasa el turno al siguiente jugador.

X comprenderá entre [ciegaGrande, su Stack]

Caso B)

La apuesta mínima para continuar en la partida es Z.

Z comprenderá entre [ciegaGrande, Stack_max]

Stack_max = Cantidad de fichas del jugador que más tiene en la mesa

Alternativa 1: hacer **Fold**, el jugador se retira de esa mano.

Alternativa 2: hacer **Call**, el jugador paga la apuesta exigida Z para continuar, y pasa el turno al siguiente jugador.

Alternativa 3: hacer **Raise**, el jugador sube la apuesta de la mesa en W.

Se debe cumplir siempre que: $W + \text{CiegaGrande} \geq Z$

El bot debe elegir la mejor alternativa en cada situación, para ello el algoritmo realizará los siguientes cálculos:

1º) calcular los datos de partida

2º) calcular la utilidad del jugador (a partir de los datos de partida)

3º) comparar la utilidad con los parámetros Flop y Raise

El cálculo de los datos de partida y el cálculo de la utilidad corresponde al apartado 4.2.1

Una vez tenemos estos 4 datos de partida necesitamos calcular la utilidad del jugador. Para ello usaremos la siguiente fórmula:

$$\text{Utilidad} = V * \text{pesoV} + P * \text{pesoP} + B * \text{pesoB} + S * \text{pesoS}$$

Utilidad es el potencial que ese jugador tiene durante el momento de tomar una decisión dada sus cartas, su posición en la mesa, etcétera.

V, P, B y S son los 4 datos de partida explicados en el apartado 2.3 , todos tendrán un valor entre 0 y 1

pesoV representa el peso que se le da al dato de partida V.

pesoP representa el peso que se le da al dato de partidaP.

pesoB representa el peso que se le da al dato de partida B.

pesoS representa el peso que se le da al dato de partida S.

El valor de pesoV, pesoP pesoB y pesoS, se irá modificando durante las pruebas.

Los 4 datos de partida corresponden a 4 características distintas del juego, a priori es difícil valorar qué datos de partida son más importantes que cuales otros.

Por tanto, como cada dato de partida tiene un peso, haremos pruebas con diferentes valores en los pesos y de ese modo intentaremos estimar cuál es el peso más acertado para cada dato de partida.

Durante las pruebas pondremos a diferentes bots a competir entre ellos con diferentes pesos. El bot que mejor resultados obtenga se entenderá como el bot que mejor ajustados tiene esos pesos.

Una vez tenemos la Utilidad definida, para elegir la mejor alternativa debemos considerar dos parámetros:

Además de los datos de partida se deben tratar con otro conjunto de datos que son los parámetros. Los parámetros son los datos que el algoritmo irá modificando con el transcurso de las manos y marcará la estrategia a seguir del jugador bot.

Existen 2 parámetros, el Parámetro Raise y el Parámetro Flop.

- Parámetro Raise

El parámetro Raise indicará la frecuencia o facilidad con la que el bot realizará una subida de apuestas durante una mano.

Esto significa que cuando el parámetro Raise está muy alto, se realizarán pocas subidas de apuestas. Sin embargo cuando el parámetro Raise está bajo, se realizarán muchas subidas.

- Parámetro Flop

El parámetro Flop indicará la frecuencia o facilidad con la que el bot pagará apuestas o se retirará durante una mano.

Es decir, durante una mano el jugador puede encontrarse en una situación donde para seguir jugando necesita pagar una cierta cantidad de fichas (Call), si el parámetro Flop está muy bajo, será más posible que el bot pague (Call) y continúe jugando en la mano. Sin embargo si el

parámetro Flop está alto, será más infrecuente que haga Call y por tanto se retire de esa mano (*Flop*).

Existe una relación entre los parámetros Flop y Raise. El parámetro Raise siempre debe ser mayor que Flop, se establecerá un valor fijo X mínimo de diferencia.

Por tanto, se debe cumplir siempre que **Raise - Flop \geq X**

La alternativa elegida será la que marque la Utilidad del jugador en función del valor de los parámetros Raise y Flop.

En caso que la elección del jugador bot sea entre Raise y Check la alternativa se elegirá en función de la siguiente comparación:

1) Si la utilidad es menor que el Raise, se hará Check.

Utilidad < Raise \rightarrow "Alternativa Check"

2) Si la utilidad es mayor o igual al Raise, se hará Raise

Utilidad \geq Raise \rightarrow "Alternativa Raise"

En el caso que la elección del jugador bot sea entre Raise, Flop y Call, la alternativa elegida se elegirá de la siguiente manera:

1) Si la utilidad es menor que el parámetro Flop, se hará Flop (abandonar la mano)

Utilidad < Flop \rightarrow "Alternativa Flop"

2) Si la utilidad es mayor que el parámetro Flop y menor que el parámetro Raise, se hará Call (pagar la apuesta exigida)

Utilidad \geq Flop AND Utilidad \leq Raise \rightarrow "Alternativa Call"

3) Si la utilidad es mayor que el parámetro Raise, se subirá la apuesta (Raise)

Utilidad > Raise → “Alternativa Raise”

Para entender mejor esta política, ilustraremos con un ejemplo 4 casos distintos a continuación.

Imaginemos que un jugador está jugando una mano y se encuentra en el caso que para seguir jugando necesita pagar la apuesta mínima de la mesa.

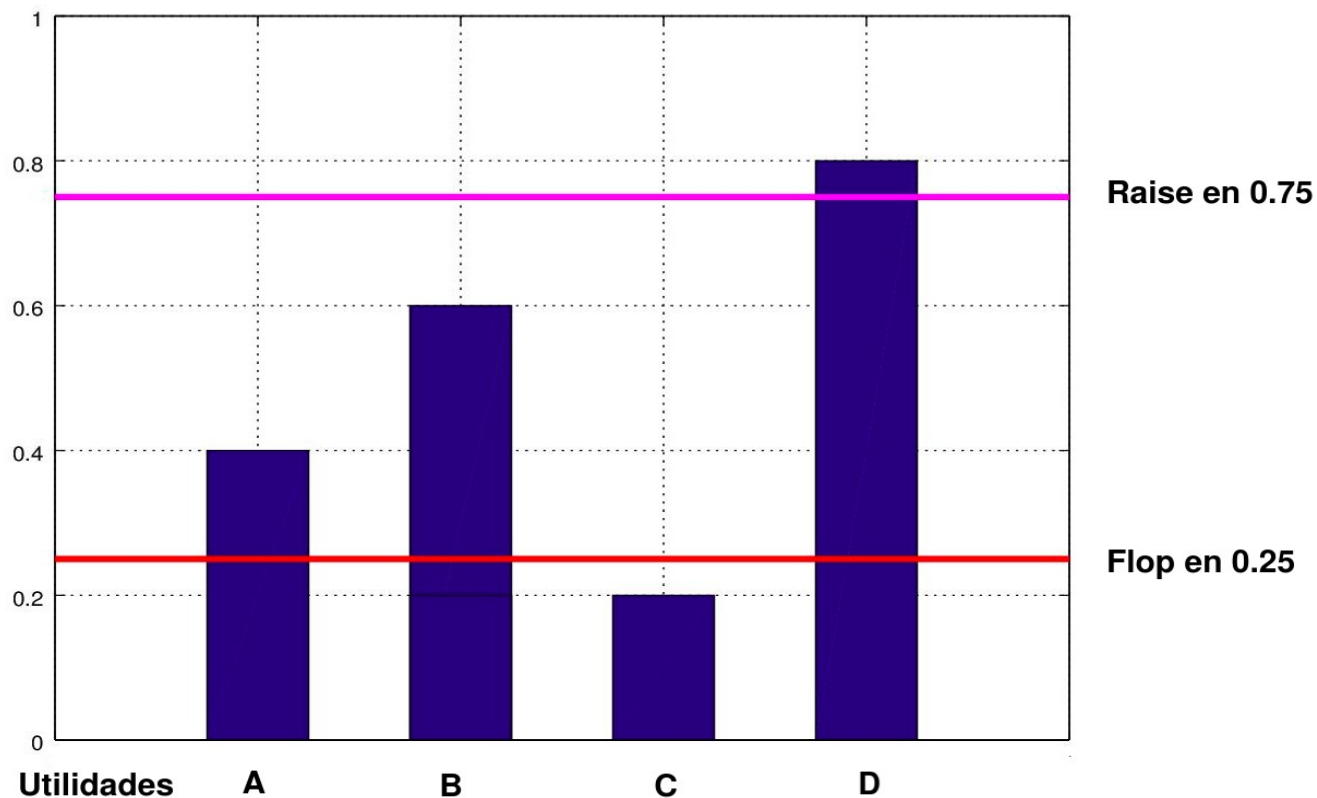
Imaginemos que el parámetro Raise y el parámetro Flop que el jugador tiene configurados se encuentran en 0.75 y 0.25 respectivamente. Según la utilidad del jugador, deberá comparar sus datos para tomar una decisión. Imaginemos 4 utilidades distintas:

Utilidad(A) = 0.4;

Utilidad(B) = 0.6;

Utilidad(C) = 0.2;

Utilidad(D) = 0.8;



(Imagen 14) Proceso de elección de alternativa: comparación de las utilidades y los parámetros

Podemos observar que las **Utilidades A y B** están por encima del parámetro del Flop y por debajo del Raise, eso significa que en caso de tener esas utilidades el bot sí pagaría la apuesta para seguir jugando, **haríamos CALL**.

Con la **Utilidad C** el bot se retiraría de la mano, **haríamos FOLD**, ya que el potencial de ese bot (su utilidad en ese momento) no llega al requerimiento mínimo del parámetro Flop.

La **utilidad D** no solamente sobrepasa el parámetro Flop, sino que también sobrepasa el parámetro Raise. Eso significa que el bot subiría la apuesta, es decir **haría RAISE**.

La cantidad precisa de subida se calculará en función de lo que la utilidad sobrepase el parámetro Raise. Cuánto más sobre pase al parámetro, mayor será la apuesta.

4.3 Aprendizaje

4.3.1 Introducción

Los resultados de las manos son los que determinarán si la estrategia que el algoritmo lleva es buena o mala. Y en caso de ser mala, el algoritmo analizará si cambia los parámetros y como los cambia.

La modificación de parámetros no se realiza en cada mano, es decir, el algoritmo esperará un conjunto definido de X manos para decidir si cambia o no cambia la estrategia, que como ya se ha dicho, modificar la estrategia significa que modificar los parámetros Raise y Flop.

La razón de esperar a jugar un conjunto grande de partidas es que es imposible juzgar si una estrategia es buena o mala solamente con una muestra de jugadas pequeña. Se recuerda que el poker es un juego donde hay un factor de incertidumbre, no solo hay desconocimiento de las cartas del adversario sino que hay aleatoriedad en las cartas que puedan aparecer durante una mano.

Por tanto para eliminar el factor “suerte” del algoritmo, o al menos reducir su influencia, se debe mantener una estrategia definida durante un conjunto X de manos y así posteriormente analizar con más criterio si la estrategia está funcionando o no.

En el mundo del Poker, la estrategia de juego en una partida de se puede resumir en ser más arriesgado o ser más conservador. Tradicionalmente el estilo de jugar en jugadores humanos y profesionales del poker suele consistir en ser paciente, no arriesgar muchas fichas hasta que se tenga una buena oportunidad y cuando llegue dicha oportunidad, arriesgar para intentar ganar el bote de la mesa.

Debemos dejar claro que el objetivo y donde se pone más énfasis es en “no perder” fichas (teoría de la mínima pérdida) y no tanto en ganar fichas. Esto significa que el algoritmo se centra en intentar conseguir una estrategia estable que no genere pérdidas.

Para modificar la estrategia deberemos seguir los siguientes pasos:

1) Recopilar datos del juego durante N manos

2) evaluar si se han ganado o perdido fichas durante esas N manos

3) modificar o no la estrategia, lo cual depende del apartado de evaluación:

- 3 a) en caso que se hayan ganado más fichas que perdido, no se modificará la estrategia. Es decir, no se modificarán los parámetros.

- 3 b) en caso que se hayan perdido más fichas que ganado, se modificará la estrategia.

4.3.2 Recopilación de datos

Tras finalizar cada mano el algoritmo recojerá las alternativas tomadas por un bot en las diferentes fases del juego. Es decir qué alternativa tomó en el Preflop, en el Flop, en el Turn y en el River.. Asociará esas decisiones tomadas en cada fase con el resultado final de esa mano, es decir, con la cantidad de fichas que ha perdido o ha ganado el jugador. Toda esa información se recopilará para un posterior análisis.

A lo largo de N número de manos, el algoritmo habrá recogido los N vectores de datos correspondientes a las N partidas. Y entonces comenzará la evaluación de los resultados.

Ilustramos esta recolección con un ejemplo:

Durante una mano el bot B paga 50 fichas por la ciega pequeña en la fase de preflop, es decir, que hace Call, después hace Check en la fase de Flop y en la fase del Turn se retira, osea hace Fold. (hacer Check es equivalente a hacer un Call de coste 0)

El vector de información quedaría así:

Preflop	Flop	Turn	River	Resultado
Call	Call	Fold	-	-50

Después de jugar 4 manos, se podrían obtener los siguientes datos:

Preflop	Flop	Turn	River	Resultado
Call	Call	Fold	-	-50
Call	Raise	Raise	Call	+150
Fold	-	-	-	0
Call	Call	Call	Call	-150

4.3.3 Evaluación de datos recopilados

Después de recoger los resultados durante N manos, pasamos a analizar como están siendo estos. Este paso es muy sencillo, pues solamente tenemos que analizar si estamos ganando o perdiendo con la estrategia actual. Para eso realizaremos un sumatorio de todas las anteriores partidas.

Siguiendo el ejemplo anterior, si hiciéramos la evaluación después de 4 manos, obtendríamos los siguientes resultados.

Preflop	Flop	Turn	River	Resultado
Call	Call	Fold	-	-50
Call	Call	Raise	Call	+150
Fold	-	-	-	0
Call	Call	Call	Call	-150
				-50

Podemos observar que tenemos un saldo de -50 fichas tras las N manos (en este caso 4 manos).

Cuando el saldo sea positivo, se mantendrá la misma estrategia, esto es, no se modificarán los parámetros Raise y Flop.

Cuando el saldo sea negativo, sí se modificará la estrategia. Lo cual se explica en el apartado siguiente.

4.3.4 Modificación de la estrategia

Siguiendo la filosofía de la mínima pérdida, se intenta buscar donde el algoritmo está perdiendo más fichas. Para ello se analizan las 4 acciones, Call, Raise, Check y Flop, en las 4 fases (preflop, flop, turn y river) y se analiza si están generando pérdidas o ganancias.

Hay que tener presente que la alternativa Fold siempre va a dar un saldo negativo, ya que cuando se hace Fold, sea en la fase que sea, el jugador se retira de la mano y pierde lo ya apostado hasta el momento.

Hay que tener en cuenta también que cuando se hace Check significa que no se sube la apuesta, es decir, es como no hacer Raise, o hacer un Call de coste 0.

Por tanto las 2 alternativas a estudiar y sobre las que trabajaremos son Raise y Call.

Para entender mejor este concepto, se pone un ejemplo de los resultados que se podrían obtener después de haber recopilado 100 manos seguidas. Se obtendría una tabla como la siguiente:

Fase Preflop

Alternativa	Nº de veces realizada	Saldo final en esa mano
Fold	10	-100
Call	70	+100
Raise	5	-300

Esto significa por ejemplo , que en todas las manos donde en la fase de Preflop se ha hecho Fold, se han acumulado un saldo de 100 fichas perdidas.

En todas las manos donde se hizo Call en la fase de Preflop, se ha acumulado un saldo de 100 fichas ganadas.

Y así sucesivamente.

El saldo total de ese ejemplo es negativo, por tanto tendríamos que modificar parámetros. Eso lo haremos según las pautas que explicamos a continuación.

Si el Raise ha acumulado un balance de saldo positivo durante el global de partidas, de modificará los parámetros de la siguiente manera:

$$\text{parámetroRaise} = \text{parámetroRaise} - \mu$$

Si Raise por contra devolvió un saldo negativo, se modificará el parámetro raise de la siguiente manera:

$$\text{parámetroRaise} = \text{parámetroRaise} + \mu$$

El valor de μ será objeto de estudio en las pruebas, es decir se buscará el μ más óptimo.

Se seguirá la misma política en todas las fases según la siguiente tabla:

Alternativa	Resultado	Parámetro raise	Parámetro fold
Raise	Positivo	- μ	
	Negativo	+ μ	
Call	Positivo		- μ
	Negativo		+ μ

Durante las pruebas, se asignará un μ diferente a cada una de las fases y se experimentará donde se obtienen los mejores resultados.

5 Implementación y codificación

Para desarrollar el algoritmo crearemos un entorno de trabajo con las siguientes herramientas:

- Portátil Lenovo B-50
- Sistema operativo: Linux Ubuntu
- Editor de textos: Sublime Text
- Lenguaje de programación: Python
- Ejecución del programa: Shell de Linux

Crearemos un software totalmente modular donde diferenciaremos el archivo principal de ejecución con todas las funciones que estaran agrupadas por temática en archivos diferentes.

Los archivos de salida los imprimiremos en el mismo directorio donde se ejecuta el programa.

Nuestro código se divide en varios módulos que a continuación se detallan:

main.py

- contiene la ejecución principal del programa, desde éste archivo se configuran el número de partidas y la configuración inicial del algoritmo.

Functions.py

- Contiene todas las funciones básicas del sistema.

Ranking2.csv

- Contiene la valoración de las cartas preflop de manera estructurada. Este archivo es solo de consulta para contruir la valoración de las cartas

mostrarCartas.py

- Consiste en el desarrollo de una función que muestra por pantalla (por la consola de la Shell de linux) las cartas que se le pidan

escaleraColor.py

- Consiste en el desarrollo de una funcion que dado un conjunto de cartas comprueba si existe en dichas cartas alguna combinación que valide una escalera de color.

escaleraColor.py

- Consiste en el desarrollo de una funcion que dado un conjunto de cartas comprueba si existe en dichas cartas alguna combinación que valide una escalera de color.

poker.py

- Consiste en el desarrollo de una funcion que dado un conjunto de cartas comprueba si existe en dichas cartas alguna combinación de jugada de Poker, es decir, cuatro cartas del mismo número.

full.py

- Consiste en el desarrollo de una funcion que dado un conjunto de cartas comprueba si existe en dichas cartas alguna combinación de jugada de Full, es decir, un trío y una pareja a la vez.

color.py

- Consiste en el desarrollo de una función que dado un conjunto de cartas comprueba si existe en dichas cartas alguna combinación de jugada de Color, es decir, cinco cartas del mismo palo

escalera.py

- Consiste en el desarrollo de una función que dado un conjunto de cartas comprueba si existe en dichas cartas alguna combinación de jugada de escalera, es decir, cinco cartas consecutivas en número (3,4,5,6 y 7 por ejemplo).

trio.py

- Consiste en el desarrollo de una función que dado un conjunto de cartas comprueba si existe en dichas cartas alguna combinación de jugada de Trío, es decir, tres cartas del mismo número.

doblePareja.py

- Consiste en el desarrollo de una función que dado un conjunto de cartas comprueba si existe en dichas cartas alguna combinación de jugada de Doble Pareja, es decir, dos parejas

pareja.py

- Consiste en el desarrollo de una función que dado un conjunto de cartas comprueba si existe en dichas cartas alguna combinación de jugada de pareja, es decir, dos cartas del mismo número.

cartaAlta.py

- Consiste en el desarrollo de una función que dado un conjunto de cartas comprueba cuáles son las 5 cartas de más valor.

6 Evaluación, pruebas y resultado

6.1 Introducción

El objetivo de las pruebas es encontrar la mejor configuración posible del algoritmo que genere el jugador bot más competente posible jugando al Poker en su variante *Texas Hold'em Sin límite*.

Una vez conseguida encontrada esa mejor configuración, se dispondrá a jugar un jugador humano conocedor del juego del Poker contra un jugador bot que tenga esa mejor configuración encontrada tras las pruebas.

Por tanto, en las pruebas se diferencian 2 actividades distintas:

1- hacer pruebas de juego entre jugadores bots, donde todos usan el mismo algoritmo pero con configuraciones diferentes de parámetros. Y analizar cual es la configuración más competente. Lo cual será considerado como la parte de entrenamiento.

2- Una vez hayada la mejor configuración posible, enfrentaremos a un jugador humano conocedor del juego del Poker contra un jugador bot que use el algoritmo con su mejor configuración. Está será la parte de Test.

Las pruebas de realizarán por Torneos que funcionarán de la siguiente manera:

- Todos los jugadores comenzarán con el mismo número de fichas
- La mesa (la plataformade juego) no repondrá fichas a ningún jugador, el jugador que pierda N fichas en una mano, comenzará la mano siguiente con N fichas menos.
- Cuando un jugador quede sin fichas, quedará eliminado del Torneo (su registro histórico de jugadas quedará almacenado para su posterior estudio).
- No habrá ningún tipo de coaliciones. Todos buscarán ganar el torneo y jugarán de manera individual y racional.
- Cuando solo quede un jugador, éste quedará como el vencedor de dicho Torneo

6.2 Entrenamiento

La fase de entrenamiento consiste en enfrentar a jugadores bots con diferente configuración entre ellos e durante un conjunto de Torneos seguidos y analizar los resultados finales para ver quien es el mejor jugador bot, o mejor dicho, cual es la mejor configuración de parámetros.

Para realizar el entrenamiento debemos definir que parámetros vamos a probar:

Valores a los pesos: (para la fórmula de la utilidad)

- **pesoV**

- **pesoP**

- **pesoB**

- **pesoS**

Y diferentes valores a Mu: (para la configuración de los parámetros)

- **μ**

Por otra parte, los **parámetros Raise y Fold** se irán ajustando automáticamente durante las partidas, de ese modo, también será parte del análisis evaluar cual es la mejor configuración de los parámetros Raise y Fold.

Prueba 1

Con una misma configuración inicial de parámetros, enfrentamos a 5 bots y observamos como quedan sus parámetros Raise y Fold:

Número de Jugadores: 5

Número de Torneos jugados : 400

Fichas iniciales a cada jugador: 1000

Valores iniciales:

Parámetro Raise	Parámetro Fold	Peso V	Peso P	Peso B	Peso S	Mu (μ)
0,6	0,25	0,5	0,15	0,1	0,25	0,05

Resultados:

	Torneos Ganados	Balance de Saldo Final (en miles)	Parámetro Raise final	Parámetro Fold final
Jugador 1	202	+610	0,55	0,3
Jugador 2	3	-385	0,95	0,9
Jugador 3	2	-390	0,95	0,9
Jugador 4	188	+540	0,6	0,25
Jugador 5	5	-375	0,95	0,9

Nota: Balance en Saldo Final, significa la diferencia de cantidad de fichas ganadas menos las perdidas, de tal modo que si un jugador ha ganado a lo largo de todos los torneos 10 mil fichas y ha perdido 11 mil fichas, tendrá un balance final de menos mil fichas.

Evaluación:

- Los jugadores que han mantenido una configuración del parámetro Raise de 0,6 o 0,55 tienen mucho mejor resultado que el resto.
- Los jugadores que mantenían una configuración del parámetro Fold de 0,25 o 0,3 tienen mejor resultado que el resto.
- Los jugadores con peores resultados han aumentado al máximo sus parámetros Raise y Fold.

A partir de estos datos se recordará la configuración de más éxito, que es la que posee un Parámetro Raise de 0,55 y un parámetro Fold de 0,3.

Prueba 2

En esta ocasión, se probarán diferentes valores de μ , todos los jugadores comenzarán con los mismos parámetros de Raise y Fold, que coincidirán con los mejores obtenidos en la prueba 1.

Número de Jugadores: 5

Número de Torneos jugados : 400

Fichas iniciales a cada jugador: 1000

Valores iniciales:

Jugador	Parámetro Raise	Parámetro Fold	Peso V	Peso P	Peso B	Peso S	Mu (μ)
1	0,55	0,3	0,5	0,15	0,1	0,25	0,1
2	0,55	0,3	0,5	0,15	0,1	0,25	0,08
3	0,55	0,3	0,5	0,15	0,1	0,25	0,06
4	0,55	0,3	0,5	0,15	0,1	0,25	0,04
5	0,55	0,3	0,5	0,15	0,1	0,25	0,02

Resultados:

	Torneos Ganados	Balance de Saldo Final (en miles)	Parámetro Raise final	Parámetro Fold final
Jugador 1	2	-390	0,95	0,9
Jugador 2	11	-345	0,95	0,94
Jugador 3	5	-375	0,97	0,96
Jugador 4	187	535	0,43	0,42
Jugador 5	195	575	0,55	0,3

Evaluación:

Podemos observar que el jugador con más victorias es el jugador 5, el cual tenía asignado el menor valor μ de todos, 0.02. Observamos que sus parámetros Raise y Fold se han mantenido como al principio. El jugador 4 también ha tenido buenos resultados aunque sí ha modificado ligeramente los parámetros Raise y Fold dejándolos casi idénticos: 0.43 y 0.42.

Con esta segunda prueba se percibe que los mejores resultados suelen venir con los parámetros Raise y Fold en 0.55y 0.3.

Prueba 3

En esta ocasión, se probarán diferentes valores de los pesos. Cada jugador tendrá los pesos repartidos de forma diferente, pero el sumatorio de los pesos siempre será 1.

Como novedad en esta prueba, los parámetros Raise y Fold estarán fijos durante toda la prueba, de ese modo se podrá comprobar con mayor equidad cual reparto de pesos es el más acertado.

En esta prueba el foco se pondrá en el peso V, a priori el más importante de todos. Cada jugador tendrá un pesoV diferente, y el resto de pesos estarán equilibrados.

El valor de μ (Mu) es irrelevante en esta prueba pues hemos fijado los parámetros Raise y Fold de manera constante.

Número de Jugadores: 5

Número de Tornos jugados: 400

Fichas iniciales a cada jugador: 1000

Valores iniciales:

Jugador	Parámetro Raise	Parámetro Fold	Peso V	Peso P	Peso B	Peso S
1	0,55	0,3	0,7	0,1	0,1	0,1
2	0,55	0,3	0,6	0,1	0,15	0,15
3	0,55	0,3	0,5	0,15	0,15	0,2
4	0,55	0,3	0,4	0,2	0,2	0,2
5	0,55	0,3	0,3	0,2	0,25	0,25

Resultados:

	Torneos Ganados	Balance de Saldo Final (en miles)	Parámetro Raise final	Parámetro Fold final
Jugador 1	91	55	0,55	0,3
Jugador 2	93	65	0,55	0,3
Jugador 3	75	-25	0,55	0,3
Jugador 4	66	-70	0,55	0,3
Jugador 5	75	-25	0,55	0,3

Evaluación:

Podemos observar que el mejor jugador ha sido el que tenía el segundo mayor valor de peso $V:0,6$, obtuvo ligeramente un poco más éxito que el jugador 1 que tenía un mayor valor de peso $V: 0,7$. Por tanto determinaremos de ahora en adelante que probablemente el mejor peso V está cercano al 0.6

Por otro lado nos fijamos que el reparto de triunfos en el torneo ha estado mucho más equitativo que en las anteriores ocasiones. La diferencia ha sido el valor constante de los parámetros Raise y Fold.

Prueba 4

En esta ocasión, se continuará con la línea de la prueba 3. Los parámetros Raise y Fold continuarán constantes y en esta prueba se pondrá el foco en el peso P.

El valor del peso V será el hallado en la prueba anterior como el que nos dio mejor resultado.

Cada jugador tendrá un peso P diferente, y el resto de pesos estarán equilibrados.

El valor de μ (Mu) es irrelevante en esta prueba pues hemos fijado los parámetros Raise y Fold de manera constante.

Número de Jugadores: 5

Número de Tornos jugados: 400

Fichas iniciales a cada jugador: 1000

Valores iniciales:

Jugador	Parámetro Raise	Parámetro Fold	Peso V	Peso P	Peso B	Peso S
1	0,55	0,3	0,6	0,3	0,05	0,05
2	0,55	0,3	0,6	0,25	0,075	0,075
3	0,55	0,3	0,6	0,2	0,1	0,1
4	0,55	0,3	0,6	0,15	0,125	0,125
5	0,55	0,3	0,6	0,1	0,15	0,15

Resultados:

	Torneos Ganados	Balance de Saldo Final (en miles)	Parámetro Raise final	Parámetro Fold final
Jugador 1	75	-25	0,55	0,3
Jugador 2	83	15	0,55	0,3
Jugador 3	73	-35	0,55	0,3
Jugador 4	73	-35	0,55	0,3
Jugador 5	96	80	0,55	0,3

Resultados;

Nos damos cuenta que el mejor resultado es el jugador 5, cuyo peso P era el menor de todos, con 0.1. Por tanto esto revela que el peso que se debe dar a P debe ser el más pequeño. (recordemos que el peso P significa el peso de la posición en la mesa)

Prueba 5

Llegamos a la quina y última prueba. En esta prueba se mantendrán todas las configuraciones ganadoras resultantes de las anteriores pruebas.

Parámetro Raise: 0,55

Parámetro Fold: 0,3

Peso V = 0,6

Peso P = 0,1

Por tanto solo resta hacer la prueba con los pesos B y S. Ambos son complementarios dado que la suma total de los pesos debe ser 1, por tanto la suma de los pesos B y S será siempre 0,3.

Número de Jugadores: 5

Número de Tornos jugados: 400

Fichas iniciales a cada jugador: 1000

Valores iniciales:

Jugador	Parámetro Raise	Parámetro Fold	Peso V	Peso P	Peso B	Peso S
1	0,55	0,3	0,6	0,1	0,25	0,05
2	0,55	0,3	0,6	0,1	0,2	0,1
3	0,55	0,3	0,6	0,1	0,15	0,15
4	0,55	0,3	0,6	0,1	0,1	0,2
5	0,55	0,3	0,6	0,1	0,05	0,25

Resultados:

	Torneos Ganados	Balance de Saldo Final (en miles)	Parámetro Raise final	Parámetro Fold final
Jugador 1	72	-40	0,55	0,3
Jugador 2	96	80	0,55	0,3
Jugador 3	89	45	0,55	0,3
Jugador 4	67	-65	0,55	0,3
Jugador 5	76	-20	0,55	0,3

Evaluación:

El mejor resultado es el del jugador 2, por tanto, teniendo en cuenta todas las anteriores pruebas, la configuración del jugador 2 es la que se indica como la mejor.

6.3 Test. Jugador bot vs Jugador humano

La fase de test consiste en una partida entre 4 bots configurados con los mejores valores que nos haya dado la fase de entrenamiento y enfrentarlos en una mesa contra un jugador humano.

La intención del test es comprobar el nivel de “inteligencia” que estos bots han adoptado en el juego del Poker gracias al algoritmo desarrollado. De ese modo se comprobará lo competitivo que puede ser el algoritmo desarrollado para jugar al Poker Texas Hold'em sin límite.

La mejor configuración que hemos conseguido en las pruebas es la siguiente:

Parámetro Raise: 0,55

Parámetro Fold: 0,3

Peso V = 0,6

Peso P = 0.1

Peso B = 0,2

Peso S = 0,1

A continuación se enfrenta en una misma mesa a un jugador humano contra 4 jugadores bots con la configuración previamente descrita.

Número de Jugadores: 5

Número de Tornos jugados: 10

Fichas iniciales a cada jugador: 1000

Resultados:

	Torneos Ganados	Balance de Saldo Final (en miles)
Jugador 1 BOT	1	-5
Jugador 2 BOT	1	-5
Jugador 3 BOT	1	-5
Jugador 4 BOT	1	--5
Jugador 5 HUMANO	6	+20

Evaluación:

A pesar que el humano no ha ganado todas las partidas, la diferencia entre el jugador bot y el jugador humano es bastante considerable.

El jugador Bot es muy previsible y además paga con frecuencia cualquier apuesta que se le presente, de ese modo el humano ha tenido una táctica fácil, esta es, esperar a tener buenas cartas y entonces hacer una gran subida, los jugadores bots con frecuencia han pagado esa gran subida y normalmente han perdido contra el humano.

7 Conclusiones Finales

Las conclusiones que sacamos de este proyecto con estos resultados son las siguientes:

- el algoritmo desarrollado no es suficientemente competitivo como para vencer a un humano con conocimiento mínimo en el Poker
- El algoritmo en muchas ocasiones tiene un comportamiento previsible.
- para mejorar el algoritmo se deberían contemplar muchas variables más
- para el algoritmo no existe la posibilidad de jugadas peculiares que los humanos realizan en el Poker como “faroles” (hacer creer que tienes mejores cartas que los demás)
- El algoritmo debería ser probado en muchas más combinaciones para hacer mejores aproximaciones de los pesos.

A continuación nombramos varios de los aspectos que deberían contemplarse si se deseara mejorar el algoritmo:

- crear un perfil de los jugadores rivales, de modo que si un jugador hace faroles o juega de manera irracional, el algoritmo debería ser capaz de detectarlo y aprovecharlo a su favor
- se debería tener en cuenta factores como “la agresividad de la mesa en general”, de modo que si el algoritmo detecta que la mesa está siendo muy agresiva (es decir, se producen muchas subidas de apuestas), el algoritmo pasaría a un estado más precavido y solo apostaría cuando tuviese una probabilidad alta de victoria.
- las subidas de apuestas deberían ser no tan proporcionales. De modo que el jugador humano no intuya tan fácilmente cuál es el potencial del jugador bot.

8 Bibliografía

- [1] <http://www.monografias.com/trabajos5/teorideju/teorideju.shtml>
- [2] Ariel Rubinstein, <http://www.sinpermiso.info/textos/de-cmo-la-teora-matematica-de-los-juegos-de-estrategia-resolver-los-problemas-de-la-eurozona-y-frenar>
- [3] <http://economipedia.com/definiciones/teoria-de-juegos.html>
- [4] English Dictionary, https://en.oxforddictionaries.com/definition/game_theory
- [5] Myerson, R.B, *Game of Theory, Analysis of Conflict*. Cambridge, London 1997
- [6] Chris Stokel-Walker, http://www.bbc.com/mundo/noticias/2015/02/150220_teor%C3%ADa_de_juegos_que_es_finde_dy
- [7] Real Academia de la lengua Española, <http://rae.es>
- [8] Ferguson y Gould, 1975
- [9] Nicholson, 1997
- [10] Maddala y Miller, 1991
- [11] Fatima S., *Principles of Automated Negotiation*. 2015
- [12] <https://kelbet.es/que-es-el-poker-y-como-se-juega.html>
- [13] <https://es.wikipedia.org/wiki/P%C3%B3quer>
- [14] http://espanol.ukpokersites.com/history_poker.html

- [15] <http://www.pokerlistings.com/poker-rules-texas-holdem>
- [16] [https://en.wikipedia.org/wiki/Deep Blue versus Garry Kasparov](https://en.wikipedia.org/wiki/Deep_Blue_versus_Garry_Kasparov)
- [17] [https://en.wikipedia.org/wiki/Polaris \(poker bot\)](https://en.wikipedia.org/wiki/Polaris_(poker_bot))
- [18] <http://poker-play.srv.ualberta.ca/>
- [19] [https://en.wikipedia.org/wiki/Cepheus \(poker bot\)](https://en.wikipedia.org/wiki/Cepheus_(poker_bot))
- [20] <https://webdocs.cs.ualberta.ca/~bowling/papers/15science.pdf>
- [21] <http://www.raketherake.com/rakeback-news/2016/09/05/poker-bot-claudico-to-be-used-for-medical-treatments/>
- [22] <https://en.wikipedia.org/wiki/Claudico>
- [23] <https://en.wikipedia.org/wiki/Libratus>
- [24] <http://www.preflophands.com/>
- [25] [https://es.wikipedia.org/wiki/Problema del caballo](https://es.wikipedia.org/wiki/Problema_del_caballo)