



UNIVERSIDAD DE CHILE
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
DEPARTAMENTO DE INGENIERÍA INDUSTRIAL

BLACKJACK Y FINANZAS: EXPLORACIÓN DE ESTRATEGIAS DE APUESTAS ÓPTIMAS PARA EL JUEGO

MEMORIA PARA OPTAR AL TÍTULO DE
INGENIERO CIVIL INDUSTRIAL

Lucas Bordeu Ossa

PROFESOR GUÍA

Charles Thraves Cortés-Monroy

MIEMBROS DE LA COMISIÓN

Carlos Pulgar Arata

José Correa Haeussler

SANTIAGO DE CHILE

2024

1. Resumen Ejecutivo

Este estudio realiza un análisis exhaustivo del juego de apuestas “Blackjack”, modelando matemáticamente el juego y aplicando principios de teoría financiera. El objetivo es explorar los límites teóricos de la ventaja que un jugador con *Información Completa* puede obtener y cómo estos hallazgos pueden mejorar las estrategias de gestión de trading de activos financieros.

El análisis abstrae las variables clave de una sesión de Blackjack para modelar la optimización de las 2 estrategias que definen el comportamiento del apostador: la *política de juego* y la *política de apuestas*. La primera se refiere a las decisiones óptimas (Pedir, Plantarse, Doblar o Abrir) durante una ronda, y se puede resolver como un Proceso de Decisión de Markov (PDM) utilizando programación dinámica. Tras optimizar este algoritmo se redujo significativamente los costos computacionales respecto a trabajos previos, permitiendo encontrar la estrategia óptima y la FMP del retorno (FMP) en menos de un segundo.

La *política de apuestas* es más compleja y depende de los intereses del apostador y el número de rondas consideradas. En el presente trabajo se modeló por completo el problema de optimización de esta política como un PDM, describiendo el estado del apostador, la función de transición y una función de recompensa coherente con principios económicos. Si bien la solución general requiere de redes neuronales, se desarrolló y computó una solución simplificada bajo ciertas suposiciones.

Utilizando la FMP del retorno por ronda y la estrategia de apuestas simplificada, se exploraron 4 modelos diferentes para la estrategia de apuestas, definidos a través de formas algebraicas con parámetros libres. Estos modelos fueron optimizados para maximizar la mediana de los retornos a través de simulaciones. Se obtuvieron 4 modelos optimizados, proporcionando la distribución del retorno asociada al rendimiento de cada uno.

Este trabajo representa un avance en la solución del problema del apostador en Blackjack y en la optimización de la gestión de portafolios. El modelamiento propuesto es directamente aplicable a la gestión de trading, sugiriendo que la solución general del Blackjack podría ser útil para la resolución del problema de optimización de portafolios.

Tabla de contenido

1. Resumen Ejecutivo	i
2. Introducción	1
3. Objetivo de la Investigación	3
3.1. Objetivos Específicos	3
4. Configuración del Juego	4
4.1. Dinámica del Juego	4
4.2. Reglas y Condiciones	6
5. Primera Parte: Política de Juego	8
5.1. Modelamiento Matemático	8
5.1.1. Ronda de Juego como PDM	12
5.1.2. Retorno Esperado de la Política	13
5.1.3. Problema de Optimización	14
5.2. Cómputo de la Solución	15
5.2.1. TFG “Las Matemáticas del Blackjack”	16
5.2.2. Reformulación Eficiente del Cómputo	18
6. Segunda Parte: Políticas de Apuestas	23
6.1. Sesión de Apuestas	23
6.1.1. Secuencia de Barajas	23
6.1.2. Ecuación para el Patrimonio	24
6.1.3. Política de Apuestas	24
6.2. Modelamiento como Proceso de Decisión de Markov	26
6.2.1. Estados	26
6.2.2. Función de Transición	26
6.3. Problema de Optimización y Función de Recompensa	27
6.3.1. Problema de Optimización	27
6.3.2. Función de Recompensa	29
6.4. Solución del Problema	33
6.4.1. Política Dependiente de la Baraja	34
6.4.2. Política Parcial Óptima	37
7. Tercera Parte: Exploración de Soluciones	44
7.1. Modelos para la Política	44
7.2. Metodología	45
7.2.1. Secuencias de Barajas de Origen	45
7.2.2. Secuencias de Rondas	46
7.2.3. Optimización de los Modelos y Testeo	46

8. Resultados	49
8.1. Resultados sobre la Política de Juego	49
8.1.1. Eficiencia de Cómputo	49
8.1.2. Rentabilidad del Juego en General	51
8.2. Resultados sobre la Política de Apuestas	54
8.2.1. Apuesta Porcentual Plana	54
8.2.2. Política Parcial dependiente de la Baraja	55
8.2.3. Apuesta Plana Neta	57
8.2.4. Apuesta Compuesta	58
8.3. Distribuciones del Retorno	62
8.3.1. Problema de Optimización: ($H = 100, R = 0$)	63
8.3.2. Problema de Optimización: ($H = 100, R = 1$)	64
8.3.3. Problema de Optimización: ($H = 1000, R = 0$)	66
8.3.4. Problema de Optimización: ($H = 1000, R = 1$)	67
9. Discusión	70
10. Conclusión	73
11. Bibliografía	74
12. Anexos	75

2. Introducción

El juego de cartas “Blackjack” ha capturado la atención de millones de personas alrededor del mundo debido a su aparente simplicidad, pero subyacente complejidad. A lo largo de los años, las estrategias para aumentar las posibilidades de ganar en este juego de cartas han evolucionado desde meros consejos y trucos hasta complejos sistemas matemáticos y computacionales. Sin embargo, los principios matemáticos que subyacen al blackjack no se limitan al mundo del juego, y pueden ser fácilmente extendidos para modelar el problema de gestión de trading en el ámbito financiero.

Motivado por la curiosidad y la pasión por el Blackjack, este trabajo parte de un análisis del código que computa la “política de juego óptima” presentado en “Las matemáticas del Blackjack” [7], para luego aventurarse más allá en la optimización de la “política de apuestas”. A diferencia del problema de la política de juego, que ha sido estudiado desde hace décadas [1][2] y se considera resuelto por la comunidad [7][8][9], el problema de optimización para la política de apuestas permanece como un terreno menos investigado. Si bien algunos trabajos han estudiado el problema de manera simplificada [3][4][5], ninguno modela formal y completamente el problema de optimización para la política de apuestas.

En el presente estudio se desarrolla un planteamiento matemático riguroso para el problema de optimización de ambas políticas, computando eficientemente la política de juego óptima y la Función de Masa de Probabilidad del retorno asociado a una ronda del juego, a través de programación dinámica y análisis combinatorial. Posteriormente, se resuelve una forma simplificada para el problema de la política de apuestas a partir de las ideas de J.L.Kelly [6] y las FMP asociadas a cada ronda, computando una política óptima simplificada y brindando una aproximación para la distribución de retornos asociada. Finalmente, se proponen modelos algebraicos más complejos para la política de apuestas dotados de parámetros libres, que tras ser ajustados con técnicas de optimización estocástica basadas en simulaciones, permiten explorar las propiedades asociadas al problema de optimización general para la política, analizando los modelos optimizados resultantes y las distribuciones de retorno asociadas.

La obtención explícita de políticas de apuestas ganadoras y superiores a las conocidas tiene un impacto directo en la industria, donde se ha visto históricamente cómo los Casinos han realizado modificaciones al juego ante la aparición de nuevos mecanismos para mejorar la ventaja del jugador. Con la aparición del libro de Thorp, por ejemplo, los Casinos reaccionaron añadiendo mazos a la baraja y variando el “punto de corte”, con la idea de contrarrestar las posibles estrategias del jugador [10]. Si bien las políticas halladas en este trabajo solo pueden ser ejecutadas con un computador, fácilmente pueden ser adaptadas para ser viables en Casinos de manera presencial. Esto sin duda resulta relevante para una industria de millones de dólares, que no puede darse el lujo de no reaccionar ante la aparición de nuevas estrategias ventajosas para los jugadores, pero que tampoco puede restringir en exceso las condiciones

del juego, espantando a los jugadores.

Sin embargo, el aspecto más relevante de este trabajo no está asociado al juego. Como se mencionó, el problema general de la política de apuestas en el Blackjack es matemáticamente análogo al problema de gestión de trading de activos financieros. La variable clave que diferencia ambos problemas es el retorno de cada periodo. Mientras que en el Backjack la FMP de dicha variable es conocida de manera exacta en cada ronda de apuestas a partir de la composición de la baraja, el retorno asociado a la variación del valor de un activo en un determinado periodo solo puede modelarse de manera aproximada a partir de variables del mundo real, y siempre conlleva un error. Se espera que la resolución del problema para el Blackjack, donde la FMP del retorno por periodo es conocida, sirva como referente para futuras investigaciones asociadas a la gestión de trading, considerando que el modelamiento del retorno por periodo para activos financieros será cada vez más preciso, en particular debido al crecimiento exponencial que está experimentando la inteligencia artificial, los datos, y la tecnología en general.

En resumen, esta tesis se propone estudiar las políticas de apuesta óptimas para el Blackjack desde una perspectiva financiera, explorando soluciones simplificadas y estableciendo un modelamiento riguroso para el problema general. Además, se espera que los resultados obtenidos y las ideas propuestas en este trabajo sirvan como referente para futuras investigaciones, ayudando a profundizar en la comprensión del problema de optimización de gestión financiera.

3. Objetivo de la Investigación

“El objetivo de esta investigación es estudiar las políticas de apuesta óptimas para el Blackjack, mediante un modelamiento matemático riguroso para el problema de optimización y la exploración de soluciones simplificadas. Se busca superar el rendimiento logrado por las estrategias estudiadas hasta la fecha, y que los resultados de esta investigación sirvan como base para hallar la solución general del problema y para enriquecer la comprensión de la gestión de portafolios financieros. Este objetivo también es relevante para la industria del juego, donde la aparición de nuevas estrategias rentables para los jugadores tiene un impacto directo en las ganancias de los Casinos.”

3.1. Objetivos Específicos

1. Modelar completamente el problema de optimización para la política de juego.
2. Desarrollar un programa que calcule la política de juego óptima y la FMP para el retorno de una ronda en tiempo real (menos de 5 segundos).
3. Modelar el problema de optimización para la política de apuestas.
4. Encontrar la solución teórica y computacionalmente para un caso simplificado del problema, que incorpora una política de apuestas con información parcial.
5. Proponer, optimizar y evaluar el rendimiento de modelos algebraicos para la política de apuestas.

4. Configuración del Juego

Existen múltiples aspectos que definen totalmente una sesión de juego de Blackjack, por lo cual se ha de ser preciso a la hora de referirse al juego como objeto de estudio. En este apartado se describe la “*dinámica del juego*” y las “*reglas y condiciones*” que se asumirá que *rigen durante toda la investigación*. La dinámica del juego se refiere a una descripción de en qué consiste y cómo se desarrolla el juego. Las reglas y condiciones hacen alusión a un conjunto específico de reglas y condiciones consideradas en particular para esta investigación (hay muchas variaciones).

4.1. Dinámica del Juego

El Blackjack es un juego de cartas que se juega en una mesa de casino o a través de plataformas online, donde múltiples jugadores pueden participar simultáneamente, aunque cada uno compite individualmente contra la casa, representada por el crupier. La esencia del juego radica en la estrategia y toma de decisiones basadas en el valor de las cartas recibidas, con el objetivo de obtener un puntaje cercano o igual a 21, sin excederse.

Inicio del Juego y Apuestas:

Antes de que la ronda comience, cada jugador en la mesa realiza una apuesta inicial. Esta apuesta es el requisito para que el jugador sea considerado en la ronda que está por iniciar. Las apuestas se colocan en un espacio designado frente a cada jugador.

Reparto de Cartas:

Una vez realizadas las apuestas, el crupier procede a repartir las cartas. Cada jugador recibe dos cartas boca arriba, mientras que el crupier recibe una carta boca arriba y, dependiendo de la variante del juego, puede recibir otra carta boca abajo (conocida como “la carta oculta”).

Valor de las Cartas:

- Cartas numéricas (2 a 10): Valen su valor facial.
- Monos (J, Q, K): Valen 10 puntos.
- Ases: Tienen un valor dual, pudiendo valer 11 puntos o 1 punto, dependiendo de la conveniencia para la mano del jugador. Un As se cuenta como 11 a menos que tal valor llevara la mano a superar los 21 puntos, en cuyo caso vale 1.

Desarrollo de la Ronda:

Tras el reparto inicial de cartas, los jugadores evalúan sus manos y proceden con su jugada,

teniendo las siguientes opciones:

- Pedir: Solicitar una carta adicional para intentar acercarse a 21.
- Plantarse: Mantener la mano actual, sin recibir cartas adicionales.
- Doblar: Duplicar la apuesta inicial a cambio de recibir solo una carta más.
- Abrir: Si las dos cartas iniciales tienen el mismo valor, el jugador puede dividir las en dos manos separadas, doblando la apuesta. Se reparte una carta adicional a cada nueva mano, y se juegan independientemente.
- Rendirse: En algunas variantes, el jugador puede optar por rendirse tras recibir las dos primeras cartas, recuperando la mitad de su apuesta.

Turno del Crupier:

Una vez que todos los jugadores han completado sus manos, el crupier juega su mano siguiendo reglas fijas: debe pedir hasta alcanzar un mínimo de 17 puntos. En algunas variantes, si el crupier tiene un “soft 17” (una mano que incluye un As valorado en 11 puntos), debe pedir otra carta.

Determinación del Ganador:

Una vez el crupier ha terminado de recibir cartas, finaliza la ronda y se determina quien fue el ganador:

- Blackjack: Si un jugador recibe un As y una carta de valor 10 como las dos primeras cartas, esto se llama un “Blackjack”, que es la mejor mano posible y generalmente paga más que una victoria regular, típicamente 3 a 2.
- Si el jugador supera al crupier sin pasarse de 21, gana la ronda y recibe un pago 1 a 1 sobre su apuesta.
- Si el crupier se pasa de 21 y el jugador no, el jugador gana.
- En caso de empate, la apuesta del jugador se devuelve sin ganancia adicional.
- Si el crupier tiene una mano superior sin pasarse de 21, el jugador pierde su apuesta.

El Blackjack es un juego que no solo se basa en la suerte de las cartas repartidas, sino también en las decisiones estratégicas del jugador. La habilidad para decidir cuándo pedir, plantarse, doblar o abrir, tomando en cuenta la carta visible del crupier y las cartas que ya se han jugado, puede influir significativamente en el éxito del jugador.

4.2. Reglas y Condiciones

Dependiendo de la variante del juego, se establecen distintas reglas y condiciones específicas respecto al desarrollo del juego y las acciones que puede realizar el jugador o el crupier y cuándo pueden realizarlas. Para esta investigación en particular, se trabajó con el siguiente set de reglas:

Conjunto de Reglas \mathfrak{R} :

1. **Baraja:** La baraja a partir de la cual se reparten cartas se compone de 8 mazos, donde cada mazo se compone a su vez de 52 cartas.
2. **Punto de Corte:** Corresponde al porcentaje de cartas de la baraja que se juegan antes de reiniciarla a su estado completo. Para este estudio se consideró un punto de corte de 75 %.
3. **Crupier:** El crupier siempre se planta siempre si su puntaje es igual o mayor a 17. El crupier no revisa si tiene Blackjack cuando su carta es mono o as.
4. **Doblar:** Se permite doblar con cualquier mano compuesta de 2 cartas. No es posible doblar luego de abrir. Luego de doblar la apuesta, el jugador recibe una única carta.
5. **Abrir:** Se permite abrir cualquier mano compuesta por 2 cartas, siempre y cuando estas sean iguales. Solo se puede abrir una vez por ronda. Si se abre un par de ases, el jugador recibe una única carta para cada as, y si se obtiene blackjack en alguna de ambas manos, el premio es 1 a 1.
6. **Pedir o plantarse:** Se permite pedir o plantarse en cualquier momento, siempre que se respeten las otras reglas.
7. **Blackjack:** Paga 3 a 2.
8. **Rendirse y seguros:** No se permite.

La dinámica del juego en conjunto con las reglas \mathfrak{R} definen inequívocamente un juego de Blackjack. Por lo tanto, si se establecieran estrategias de juego deterministas para cada jugador de la mesa, se podría en principio deducir la probabilidad de cada escenario posible que se puede dar durante la ronda, considerando la composición de la baraja al inicio de la ronda, que codifica las probabilidades de que se reparta cada carta. Es decir, si el jugador estuviese en conocimiento de la estrategia que sigue cada jugador de la mesa, de la composición exacta de la baraja en cada momento, de las reglas del juego y de su propia estrategia de juego, en teoría estaría en condiciones de conocer la Función de Masa de Probabilidad (FMP) del retorno de la ronda previo a que esta comience. Este escenario se conoce como “Información Completa”, pues se cuenta con toda la información relevante del juego.

Sin embargo, este es un escenario improbable, principalmente por la presencia de jugadores adicionales en la mesa, pues la posibilidad de conocer su estrategia es virtualmente nula. Incluso es posible que no sigan estrategias concretas, siguiendo a veces una “corazonada” o

factores aleatorios. Pero ¿qué ocurre cuando no hay más jugadores en la mesa?, esta es una simplificación poderosa del problema, pues ahora el jugador solo necesita conocer las reglas del juego y la composición de la baraja para estar en condiciones, al menos en teoría, de conocer de manera exacta la FMP de sus retornos a partir de su propia política de juego durante la ronda. Por lo tanto, durante esta investigación *se asumirá la condición adicional de que el jugador juega solo contra el crupier*, y por lo tanto acceder a la información necesaria para deducir la distribución del retorno sí es posible, al encontrarnos en un caso de “Información Completa”.

***NOTA:** *Para que el jugador pueda conocer efectivamente la FMP del retorno de una ronda, es indispensable que maneje información precisa sobre las cartas que componen la baraja en cada instante. En la práctica, esto se consigue mediante un “conteo” de las cartas que se van jugando, lo que permite saber cuáles quedan en la baraja. Esta técnica, ampliamente utilizada por los jugadores de Blackjack, es legal en la mayoría de los países; sin embargo, suele estar prohibida en las mesas de juego de los casinos. Durante esta investigación, se asumirá que el jugador lleva una cuenta exacta de las cartas jugadas, conociendo así la composición de la baraja en todo momento. Si bien este supuesto va contra las normas de los casinos y supone retos significativos para el jugador en la práctica, favorece un modelamiento teórico riguroso del problema, y permite estudiar el juego con una mayor profundidad y precisión que si no se asumiera.*

5. Primera Parte: Política de Juego

5.1. Modelamiento Matemático

Supongamos entonces la *ausencia de jugadores adicionales y una configuración de reglas* \mathfrak{R} fija a lo largo del juego. Supóngase además que el jugador lleva la cuenta precisa de las cartas jugadas desde el inicio, con baraja completa, y que por lo tanto conoce la composición de la baraja justo antes de comenzar cada ronda de apuestas. A la composición de la baraja previo al comienzo de una ronda le llamaremos *baraja de origen* de la ronda, y la definiremos a través de un vector de 10 componentes b :

Baraja de Origen:

$$b \in \mathbb{R}^{10}$$

Donde b_i es la cantidad de cartas en la baraja cuyo valor en el juego es i (a excepción de los ases, que pueden tomar 2 valores), aclarando que b_1 representa la cantidad de ases y b_{10} la cantidad de combinada de “10”, “jotas”, “reinas” y “reyes”.

Ahora, dado que para comenzar una nueva ronda se debe corroborar que no se ha alcanzado el “punto de corte” de la baraja, definimos el espacio de valores que puede tomar b durante un *juego completo*, que entenderemos como una secuencia de rondas de apuestas desde que está la baraja en su estado inicial hasta que se ha alcanzado el “punto de corte”. Se define entonces el espacio de barajas de origen Ω como sigue:

Espacio Barajas de Origen:

$$\Omega_{c,m} = \left\{ b \in \mathbb{R}^{10} \mid 52m(1-c) \leq \sum_{i=1}^{10} b_i \leq 52m, b_i \leq 4m \ \forall i \leq 9, b_{10} \leq 16m \right\}$$

Con:

- $M_{c,m} = \text{card}(\Omega_{c,m}) = |\Omega_{c,m}| \in \mathbb{N}$: Cantidad de barajas de origen posibles.
- $m \in \mathbb{N} = \{1, 2, \dots\}$: Número de mazos que componen la baraja completa inicial. Donde un mazo se representa por la baraja $b = [4, 4, 4, 4, 4, 4, 4, 4, 4, 16]$.
- $c \in [0, 1]$: Punto de corte establecido para el juego. Es decir, hasta que fracción de la baraja completa inicial se procede a jugar una nueva ronda. Una vez se pasa este punto, se reinicia la baraja a su estado inicial $b = [4m, 4m, 4m, 4m, 4m, 4m, 4m, 4m, 4m, 16m]$.

Ahora bien, como el jugador está en conocimiento tanto del estado de la baraja de origen b de una ronda como del comportamiento del crupier durante la ronda, que es determinístico,

entonces es capaz de calcular la probabilidad de todos los posibles resultados de la ronda, en función de todas las secuencias aleatorias de cartas que pueden ocurrir a partir de la baraja de origen b , *condicionadas según su propio comportamiento o estrategia de juego durante la ronda*.

Podemos definir entonces 6 resultados posibles por ronda, en términos de los retornos obtenidos sobre la apuesta inicial realizada por el jugador, que dependen del tipo de decisión tomada: *perder el doble de la apuesta inicial, perder un monto igual a la apuesta inicial, empatar, ganar un monto igual a la apuesta inicial, ganar 1.5 veces la apuesta inicial (Blackjack) y ganar el doble de la apuesta inicial*. Estos 6 resultados constituyen el espacio de posibles retornos sobre la apuesta inicial en cada ronda (existen variaciones del juego donde puede haber otros retornos):

Espacio de Retornos sobre apuesta inicial:

$$\text{PagosBJ} = [-2, -1, 0, 1, 1.5, 2]$$

Entonces, si b es el estado de la baraja *previo al comienzo de la ronda*, es posible calcular el vector con las probabilidades para cada uno de estos 6 retornos de manera exacta, a partir de la dinámica propia del juego bajo un set de reglas \mathfrak{R} y una estrategia de decisión establecida para el jugador.

Antes de formalizar la función que calcula la función de masa de probabilidad (FMP) de los 6 posibles retornos para una ronda en general, resulta necesario introducir una función que determine la *política de juego* durante la ronda, y en particular aquella política que maximiza el retorno esperado de la ronda. Es decir, una función que asigne una de las decisiones Pedir, Plantarse, Abrir o Doblar, para cada *instancia* a la que se enfrenta el jugador durante una ronda, de tal manera que la esperanza del retorno final de la ronda sea máxima.

Dado que una vez el jugador ha establecido una política de juego entonces su comportamiento durante una ronda es determinista, luego dicha política en conjunto con la baraja de origen $b \in \Omega_{c,m}$ determinan por completo la probabilidad de cada posible secuencia de cartas que se pueden repartir durante la ronda. Estas secuencias las identificamos con instancias w , que a continuación definimos incorporando las variables necesarias para que representen inequívocamente los *estados* o situaciones relevantes durante una ronda.

Una *instancia* w se define completamente por las siguientes variables: la baraja de origen de la ronda b , el puntaje de la primera mano del jugador J^1 , un indicador de si la primera mano del jugador es dura (no hay ningún as valiendo “11” en la mano (0)) o blanda (existe un as cuyo valor es “11” en la mano (1)) x^1 , el número de cartas que componen la primera mano del jugador y^1 , el puntaje de la segunda mano del jugador en caso de haber decidido

“Abrir” su primera mano J^1 , un indicador de si la segunda mano del jugador es dura x^2 , el número de cartas que componen la segunda mano del jugador y^2 , el puntaje del crupier a partir de sus cartas visibles C , un indicador de si la mano del crupier es dura o blanda z , el número de cartas que componen la mano del crupier v , y la composición de la *baraja de ronda* q .

Se introducen entonces la familia de conjuntos \mathcal{W}^b , que contienen todas las instancias que pueden darse durante una ronda a partir de una baraja de origen b , de acuerdo con la dinámica propia del juego:

Instancias de Ronda:

$$\mathcal{W}^b = \{w = (J^1, x^1, y^1, J^2, x^2, y^2, C, z, v, q)\}$$

Con:

- $J^1, J^2 \in \{0, 2, 3, \dots, 30\}$.
- $x^1, x^2, z \in \{0, 1\}$.
- $y^1, y^2 \in \{0, 1, 2, \dots, 20\}$.
- $C \in \{0, 2, 3, \dots, 26\}$.
- $v \in \{0, 1, 2, \dots, 12\}$.
- $b \in \Omega_{c,m}$.
- $q \in Q^b$: Baraja de ronda. Se refiere al estado de la baraja durante una ronda de juego, con baraja de origen b .
- $Q^b = \{q = (b - k) | q_i \geq 0, k \in \Omega_{1,m}, \sum k_i \leq 51\}$: Espacio Barajas de Ronda, a partir de una baraja de origen b .

Ahora, dado que nos interesa asignar decisiones a una secuencia de instancias que ocurren durante una ronda, y el traspaso de una instancia a otra ocurre al repartirse una nueva carta durante la ronda, definimos en particular los conjuntos con las *instancias de s cartas* \mathcal{W}_s^b , que contienen todas aquellas instancias w provenientes de una baraja de origen b cuya baraja de ronda q posee s cartas menos que la de origen.

Instancias de s Cartas:

$$\mathcal{W}_s^b = \left\{ w \in \mathcal{W}^b \mid \sum_{1 \leq i \leq 10} (b - q)_i = s \right\}$$

Con:

$$w = (\dots, q)$$

$$q \in Q^b$$

A una instancia w tal que su baraja de ronda q posee s cartas menos que la de origen la anotaremos como w_s , y la llamaremos “instancia de s cartas”.

A partir de la definición de los conjuntos \mathcal{W}_s^b , se destacan 2 subconjuntos relevantes:

- \mathcal{W}_0^b : Este conjunto contiene un único elemento $w_0 = (0, 0, 0, 0, 0, 0, 0, 0, 0, q = b)$. A la instancia w_0 le llamamos *instancia de origen*, pues se define únicamente a partir de la baraja de origen de la ronda.
- \mathcal{W}_3^b : Este conjunto contiene las *instancias iniciales* de la ronda. Son aquellas instancias asociadas a cuando recién se han repartido las primeras 2 cartas al jugador y la carta del crupier, a partir de una baraja de origen b (en otras modalidades del juego el crupier recibe 2 cartas al comienzo, con la segunda carta boca abajo).

Otro tipo de instancias que serán de especial relevancia consiste en las *instancias finales*, asociadas al término de la ronda. Cuando la ronda se encuentra en estas instancias, ni el jugador ni el crupier pueden recibir más cartas, y se determina quien fue el ganador, según sus puntajes finales. Se definen entonces los conjuntos $\mathcal{W}^{F,b}$, que contienen las instancias finales asociadas a una ronda con baraja de origen b :

Instancias finales:

$$\mathcal{W}^{F,b} = \{w \in \mathcal{W}^b \mid \text{Condición}_1 \vee \text{Condición}_2 \vee \text{Condición}_3\}$$

Con:

- Condición₁: $C \geq 17$. El crupier alcanza su puntaje final.
- Condición₂: $(J^1 > 21) \wedge (J^2 > 21) \vee (J^2 = 0)$. El jugador se pasa con su primera o con ambas manos.
- Condición₃: $(J^1 = 21, y^1 = 2, C \neq 10, z = 0)$. El jugador obtiene Blackjack y el crupier no puede empatarlo.

Un aspecto clave del problema de optimización para la política de juego resulta de observar que durante una ronda de juego es común que el jugador tome varias decisiones de manera secuencial, pudiendo pedir cartas más de 1 vez, pedir y luego plantarse, abrir y luego de pedir, etc. Esto implica que el retorno asociado a la toma de una decisión en una instancia inicial w_3 , depende directamente de las decisiones asociadas a todas las instancias que pueden darse a partir de w_3 durante la ronda.

Por lo tanto, la política de juego que maximiza el retorno esperado a partir de una instancia

inicial queda determinada por la asignación de una decisión a cada una de todas las posibles instancias que pueden darse durante la ronda. De esta manera, es posible calcular la esperanza del retorno final de una ronda tras tomar una decisión en una instancia inicial, teniendo en consideración las instancias de decisión futuras que podrían ocurrir, con sus respectivas probabilidades de ocurrencia durante la ronda.

Antes de formular el problema de optimización, formalizamos primero el concepto de *política o estrategia de juego* para una ronda. Se define entonces la función *política de juego* θ , que devuelve una decisión asociada a cada instancia $w \in \mathcal{W}^b$ para cualquier baraja de origen b , de tal manera que se *respete el set de reglas* \mathfrak{R} previamente establecido:

Política de Juego:

$$\begin{aligned} \theta : \bigcup_{b \in \Omega_{c,m}} \mathcal{W}^b &\rightarrow U \\ w &\rightarrow \theta(w) \end{aligned}$$

Con:

- $U = \{\emptyset, \text{Pedir}, \text{Plantarse}, \text{Doblar}, \text{Abrir}\}$. Espacio de decisiones. Se incluye la decisión \emptyset para instancias w tal que al jugador no le corresponde tomar ninguna decisión, como aquellas asociadas al turno del crupier, o aquellas previas a las instancias iniciales.

5.1.1. Ronda de Juego como PDM

Notamos que hemos definido todas las instancias que puede ocurrir durante una ronda \mathcal{W}^b , y una política de juego θ que asigna una decisión a cada una de manera determinista. Esto significa que la probabilidad de transicionar de una instancia actual de s cartas $W_s = w_s$ a la siguiente $W_{s+1} \in \mathcal{W}_{s+1}^b$ durante una ronda, queda completamente determinada por la política de juego establecida y la instancia actual. Por lo tanto, definimos una *ronda de juego* originada desde una baraja de origen en particular $b = R$, como una secuencia H_θ^R de instancias aleatorias $\{W_s\}_{0 \leq s}$, cuya función de probabilidad de transición depende de la política establecida θ :

$$H_\theta^R = \{W_s\}_{0 \leq s}$$

1. Tal que W_s es la instancia aleatoria de s cartas, cuya FMP se define como sigue:

$$T_\theta(w_{s+1}, w_s) = \text{Prob}(W_{s+1} = w_{s+1} | W_s = w_s, \theta)$$

Con:

- $W_s \in \mathcal{W}_s^R$: Instancia de s cartas de la ronda.
- θ : Política de Juego θ , que asigna decisiones a las instancias W_s .
- $W_0 = w_0 = (0, 0, 0, 0, 0, 0, 0, 0, 0, R)$: Instancia de origen de la ronda.
- W_3 : Instancia inicial de la ronda.

* $T_\theta(w_{s+1}, w_s)$: Probabilidad de llegar a la instancia w_{s+1} a partir de una instancia w_s , sacando una carta de la baraja, bajo una política θ . Esta función define por completo la evolución de las instancias durante una ronda. Su estructura explícita se discutirá más adelante, en la sección “Cómputo de la Solución”

De esta manera, la secuencia H_θ^R define por completo la dinámica de una ronda como un *Proceso de Decisión de Markov (PDM)*, a través de la función de transición $T_\theta(w_{s+1}, w_s)$, la política de juego θ y el conjunto de instancias posible durante la ronda \mathcal{W}^R . Si se asume una política de juego establecida, luego resulta posible deducir la probabilidad de transicionar desde cualquier instancia de s cartas $W_s = w_s$ de la ronda a una instancia cualquiera $w' \in \mathcal{W}^R$, utilizando matrices de transición de Markov. A continuación, se introduce la función que calcula esta probabilidad, la cual resultará de utilidad para computar la política de juego óptima y para modelar el problema de optimización de la política de apuestas.

Se define la función $D_\theta(w', w_s)$, que devuelve la probabilidad de llegar a una instancia cualquiera $w' \in \mathcal{W}^R$, independiente de la cantidad de cartas que la define respecto a la baraja de origen R , a partir de otra instancia de s cartas w_s ocurrida en la misma ronda, siguiendo una política de juego θ :

Probabilidad de Transición Absoluta:

$$D_\theta(w', w_s) = \sum_{z \geq s} \text{Prob}(W_z = w' | W_s = w_s, \theta)$$

5.1.2. Retorno Esperado de la Política

Se destaca que llegar de una instancia a otra cualquiera *depende exclusivamente de la secuencia de cartas que se jueguen durante la ronda* H_θ^R , y consecuentemente la probabilidad por transicionar entre 2 instancias cualesquiera dependerá de la probabilidad de que se den ciertas secuencias específicas de cartas durante la ronda, bajo una política determinada. Finalmente, notando que el retorno obtenido por el jugador queda determinado por las instancias finales, y que la instancia final de una ronda es aleatoria, se tiene que el retorno por ronda para el jugador también es una variable aleatoria. Sin embargo, dado que la función de transición T_θ y la función de transición absoluta D_θ pueden ser conocidas, luego es posible en principio conocer la FMP del retorno de la ronda, a partir de la política de juego y la instancia de origen W_0 , asociada a la baraja de origen R .

Introducimos entonces el *retorno por instancia* $Y_\theta^{w_s}$, definido como el retorno final de una ronda H_θ^R a partir de una instancia de s cartas $W_s = w_s$, siguiendo una política de juego θ . Se destacan 2 formulaciones para la FMP de $Y_\theta^{w_s}$, describiendo esta función en términos recursivos y absolutos, a partir de la función T_θ y D_θ respectivamente:

Retorno por Instancia:

$$Y_\theta^{w_s} : \text{PagosBJ} \rightarrow [0, 1]$$

$$w_s \in \mathcal{W}^R$$

1. Formulación recursiva de la FMP:

$$\text{Prob}(Y_\theta^{w_s} = y) = \begin{cases} \sum_{w_{s+1} \in \mathcal{W}_{s+1}^R} T_\theta(w_{s+1}, w_s) \cdot \text{Prob}(Y_\theta^{w_{s+1}} = y) & , \text{ si } w_s \notin \mathcal{W}^{F,R} \\ f(y, w_s) & , \text{ si } w_s \in \mathcal{W}^{F,R} \end{cases}$$

2. Formulación directa de la FMP:

$$\text{Prob}(Y_\theta^{w_s} = y) = \sum_{w' \in \mathcal{W}^{F,R}} D_\theta(w', w = w_s) \cdot f(y, w')$$

* Donde $f : \text{PagosBJ} \times \mathcal{W}^{F,R} \rightarrow \{0, 1\}$, es una función que devuelve la probabilidad de obtener un retorno “y” dado que se está en una instancia final $w \in \mathcal{W}^{F,R}$. Su definición explícita resulta engorrosa por la cantidad de escenarios, pero es una función trivial pues los valores de una instancia final determinan inequívocamente si efectivamente el retorno para el jugador es “y” ($f = 1$) o no ($f = 0$).

5.1.3. Problema de Optimización

La variable $Y_\theta^{w_s}$ modela el retorno final de la ronda H_θ^R a partir de cualquier instancia de la ronda. En particular, podemos pensar en la instancia de origen w_0 , y referirnos al retorno general de la ronda, considerando todas las posibles formas en que esta se puede desarrollar. Para este caso especial de la variable $Y_\theta^{w_s}$ donde se tiene que $w_s = w_0$, anotaremos $Y_\theta^{w_s}$ como X_θ^R , y lo llamaremos *retorno por ronda*, para destacar la importancia de que nos referimos al retorno de una ronda H_θ^R desde antes que esta haya comenzado, solo a partir de la baraja de origen R y la política θ que se defina:

Retorno por Ronda:

$$X_\theta^R \Leftrightarrow Y_\theta^{w_s}$$

Finalmente, dado que el parámetro de optimalidad para la política de juego es la esperanza del retorno X_θ^R , definimos el problema de optimización para la política $\theta(w)$:

Optimización Política de Juego:

$$\theta^*(w) = \operatorname{argmax}_\theta \mathbb{E}[X_\theta^R]$$

Donde podemos expresar la esperanza a partir de la formulación recursiva y directa que establecimos para la FMP de $Y_\theta^{w_s}$.

1. Esperanza Recursiva:

$$\mathbb{E}[X_\theta^R] = \mathbb{E}[Y_\theta^{w_0}]$$

Con:

$$\mathbb{E}[Y_\theta^{w_s}] = \sum_{w_{s+1} \in \mathcal{W}_{s+1}^R} T_\theta(w_{s+1}, w_s) \cdot \mathbb{E}[Y_\theta^{w_{s+1}}]$$

2. Esperanza Directa:

$$\mathbb{E}[X_\theta^R] = \sum_{w' \in \mathcal{W}^{F,R}} D_\theta(w', w_0) \cdot \mathbb{E}[f(y, w')]$$

Solucionar este problema consiste en encontrar la política de juego óptima $\theta^*(w)$, tal que el retorno esperado $\mathbb{E}[X_{\theta=\theta^*}^R]$ de la ronda $H_{\theta=\theta^*}^R$ bajo esta política es máximo. Cuando se establezca la política óptima θ^* , hablaremos del *retorno óptimo* o *retorno optimizado por ronda*, y lo anotaremos como $X_{\theta^*}^R$ o simplemente como X^R .

Se destaca que el problema de optimización fue planteado para una ronda asociada a una baraja de origen específica R , y que por lo tanto solucionarlo solo encuentra la política de juego óptima para dicha ronda. Sin embargo, dado que cada ronda se juega independientemente y que la cantidad de barajas de origen es inmanejable en la práctica, este enfoque resulta satisfactorio y permite eventualmente jugar todas las rondas bajo la política óptima, aún cuando esta no se puede conocer en principio para cada baraja de origen posible.

5.2. Cómputo de la Solución

Como se mostró en la sección anterior, es posible calcular la FMP del retorno por ronda solo a partir de la baraja de origen R y de la política de juego θ establecida. En esta sección se explicará esquemáticamente cómo se logró computar la política óptima θ^* (exceptuando el caso de “Abrir”, que se aproximó) y la FMP del retorno óptimo $X_{\theta^*}^R$, describiendo los mecanismos utilizados para hacer el código eficiente y útil en la práctica, ya que en principio es un problema costoso computacionalmente. Como punto de referencia, se considerará el trabajo de fin de grado “Las Matemáticas del Blackjack” [7], realizado por Ricardo E. Peytaví, que brindó explícitamente un código con funcionalidades cercanas a las buscadas, pero no con la eficiencia deseada.

Sin embargo, se aclara que el cómputo eficiente de la política de juego óptima es un problema considerado informalmente resuelto. Además del trabajo de Peytaví, existen sitios web [8][9] que ofrecen programas que computan la política de juego óptima en el orden de unos pocos segundos. Si bien estos sitios no son académicamente formales, sus autores son ampliamente reconocidos en foros específicos y en la comunidad, y los resultados de sus programas han sido revisados y verificados durante más de una década (el caso de “Abrir” la mano también es abordado de forma aproximada por estos autores). Dado que no fue posible acceder al código detrás de estos programas, se procedió a realizar una formulación original de un código eficiente.

Por último, es importante destacar que computar la política de juego óptima y la FMP del retorno por ronda no es significativamente relevante para los jugadores ni para los Casinos, en cuanto esta información por sí sola tiene un impacto virtualmente nulo en la rentabilidad para los jugadores. La necesidad de computar eficientemente esta información para esta investigación radica en que permite estudiar la política de apuestas del jugador, la cual determina su comportamiento durante una secuencia de rondas. La política de apuestas que siga el jugador tiene un impacto significativo en el rendimiento de sus retornos, y estudiar políticas de apuestas optimizadas sí resulta relevante para la industria del juego, así como para la investigación de portafolios eficientes en un contexto más amplio.

5.2.1. TFG “Las Matemáticas del Blackjack”

En este trabajo, el autor aborda justamente la problemática de encontrar la política de juego óptima, y concluye con un código que calcula directamente la FMP del retorno por instancia óptimo $Y_{\theta^*}^{w_s}$ para instancias iniciales o posteriores ($w_s | s \geq 3$). En particular, el autor detalla y presenta explícitamente las siguientes funciones, cuyo argumento se ha expresado en término de las variables que definen una instancia bajo el contexto de este trabajo, $w = (J^1, x^1, y^1, J^2, x^2, y^2, C, z, v, q)$, cuando sea posible:

Funciones TFG Blackjack:

1. **CroupierProb**(C, z, v, q):

Esta función retorna la probabilidad de que el crupier obtenga un puntaje final de 17, 18, 19, 20, 21, “Blackjack” o “pasarse (+21)”, a partir de las variables de instancia “C”, “z”, “v” y “q”. El autor calcula estas probabilidades de manera recursiva, repartiendo secuencialmente cartas a la mano del crupier, transicionando de una instancia a otra con cada carta y considerando la probabilidad de dicha transición. En el Anexo A se encuentra un pseudocódigo simplificado, que representa ilustrativamente el mecanismo de esta función.

2. **JugadorDecide**(J^1, x^1, y^1, C, z, q):

Esta función calcula la mejor decisión entre “Pedir” y “Plantarse”, para la instancia

definida por las variables “ J^1 ”, “ x^1 ”, “ y^1 ”, “ C ”, “ z ” y “ q ”. La función también retorna la probabilidad de ganar o perder la ronda si se toma la decisión calculada, en conjunto con el retorno esperado correspondiente. Similar a *CrupierProb*, realiza los cálculos transicionando entre instancias recursivamente, repartiendo cartas a la mano del jugador, pero considera además la mejor decisión de cada instancia (Pedir o Plantarse) para actualizar las probabilidades en cada transición. Debido a su estructura recursiva, la función resuelve la decisión óptima con el retorno esperado asociado para todas las instancias que ramifican desde la ingresada en la función, comenzando por las instancias terminales del jugador. Este es el enfoque protocolar para resolver el problema y se basa en la programación dinámica, al resolver el problema desde el final hacia atrás, para todas las instancias que recorre la función.

En el Anexo B se presenta un pseudocódigo simplificado ilustrativo, que representa el mecanismo de esta función. La función “Auxiliar” para calcular gpS, es una función sencilla que determina las probabilidades de ganar y perder del jugador si se Planta, a partir del vector de probabilidades del crupier “ M ”, el puntaje del jugador “ J^1 ”, y las variables “ x^1 ” e “ y^1 ” (Ver trabajo original para más detalle).

3. **JugadorDobla**(J^1, x^1, y^1, C, z, q):

Esta función retorna la mejor decisión entre Pedir, Plantarse o Doblar a partir de la instancia definida, y permite obtener la FMP y el retorno esperado asociados al retorno por instancia. La función extiende la optimización que realiza *JugadorDecide*, calculando primero *JugadorDecide*(J^1, x^1, y^1, C, z, q), pero incluyendo además transiciones entre instancias adicionales asociadas a la posibilidad de Doblar para el jugador. Dado que Doblar implica recibir una única carta más, el costo computacional adicional asociado a esta decisión es despreciable, y es el cálculo de *JugadorDecide* el cuello de botella de la función. Por esta razón, no se incluye un pseudocódigo que ilustre el mecanismo de la función, al ser esencialmente análoga a *JugadorDecide*.

4. **JugDecSep**:

Esta función extiende la optimización que realiza *JugadorDecide*, incorporando la posibilidad de Abrir para el jugador. Tras optimizar la política, la función retorna la mejor decisión entre Pedir, Plantarse o Abrir a partir de la instancia definida, además de la FMP y el retorno esperado asociados al retorno por instancia.

La posibilidad de jugar 2 manos secuencialmente en una misma ronda implica un número significativamente mayor de cálculos para optimizar la política, debido principalmente a que la estrategia que se establezca para la primera mano afecta las condiciones para la optimización de la segunda. Según indica el autor, esta función “. . . puede tardar horas en calcular la conveniencia, o no, de separar un par de cartas. . .”, *por lo cual propone aproximar la política y esperanza de “Abrir” como sigue*: Supóngase que el jugador

tiene 2 cartas repetidas y que su puntaje es J , y que la puntuación que tendría si solo tuviera una de sus 2 cartas sería j_r . Luego, se ejecuta *JugadorDecide* para calcular la esperanza asociada a un puntaje j_r (pidiendo o plantándose), la cual es multiplicada por dos y utilizada como aproximación para la esperanza de Abrir. Finalmente, este valor es comparado con la esperanza asociada al puntaje J (pidiendo o plantándose), y se determina la mejor decisión. Esta aproximación “no distorsiona excesivamente los resultados”, y funciona mejor para juegos con barajas compuestas por muchos mazos y puntos de corte menos profundos.

Aspectos Relevantes para esta Investigación:

La propuesta del autor se basa en la programación dinámica, actualizando las probabilidades recursivamente, a partir de la extracción de cartas de la baraja y las decisiones óptimas de cada instancia. De esta manera, proporciona 4 funciones, que permiten deducir la política de juego óptima desde cualquier instancia inicial o posterior (las funciones solo son válidas para $C > 0$ y $n > 0$). Sin embargo, el autor destaca al final del trabajo que “. . . los costes en términos de tiempo de computación son enormes, al menos con un ordenador portátil . . .”, por lo que carecen de utilidad práctica. En particular, la decisión de “Abrir” la mano conlleva un coste computacional excepcionalmente alto respecto a las otras decisiones.

El trabajo citado permite concluir que hacer eficiente la función *JugadorDecide* permitiría deducir la política de juego óptima restringida a Pedir, Plantarse o Doblar. Además, el autor propone una aproximación razonable para estimar la esperanza de “Abrir” a partir del cómputo de *JugadorDecide*, sobre todo cuando la baraja contiene muchos mazos, como es el caso del presente estudio (recordando, se ha establecido una baraja compuesta de 8 mazos, y un punto de corte de 75 %).

5.2.2. Reformulación Eficiente del Cómputo

Ya hemos introducido el trabajo de fin de grado “Las Matemáticas del Blackjack”, que a partir de las funciones *CrupierProb*, *JugadorDecide*, *JugadorDobla* y *JugDecSep* optimiza completamente la política de juego para cualquier instancia inicial o posterior ($w_s | s \geq 3$), aunque en tiempos prohibitivos. En esta sección se propone una reformulación del código (se utiliza otro set de reglas del juego, pero esto no tiene un mayor impacto en el código) desde una perspectiva matemática y técnica, permitiendo mejorar notablemente la eficiencia de *JugadorDecide*. En particular, se identifican propiedades matemáticas ligadas al cálculo de la función *CrupierProb* que reducen notablemente la cantidad de cálculos necesarios, así como la posibilidad de optimizar los algoritmos a nivel técnico utilizando diccionarios, vectorización y paralelización.

Además, el código propuesto en este trabajo calcula de manera exacta la esperanza para “Abrir”, aunque asumiendo la utilización de la política dictada por *JugadorDecide* en cada

mano “separada” (la optimización de la política permitiendo “Abrir” no puede realizarse en tiempo real utilizando un computador estándar). Por último, la formulación propuesta extiende su funcionamiento para cualquier instancia de una ronda, en particular la instancia de origen w_0 , permitiendo deducir la FMP del retorno por ronda óptimo $X_{\theta^*}^R$.

Debido a la complejidad y grado de refinamiento de los detalles involucrados en la formulación explícita del código, *solo se dará una explicación aproximada del concepto aplicado*, acompañada de pseudocódigo ilustrativo, suficiente para comprender la idea principal de la implementación. Se han incluido en la explicación las 5 funciones programadas más relevantes para comprender la reformulación del código, en un orden que facilita la comprensión de la lógica de la implementación:

Funciones Principales:

- ProbSecsCrup
- GanarPerder
- Bayes
- JugadorDecideRapido
- JugadorAbre

A continuación, se detallan las funcionalidades y la lógica de cada función, acompañando de pseudocódigo cuando valga la pena. Se reitera que las explicaciones son aproximadas, y solo buscan explicar el concepto con mayor claridad. *El código riguroso difiere de lo presentado a continuación*, principalmente manejando correctamente todos los detalles e incorporando elementos técnicos que permiten hacer eficiente el algoritmo. La comparación de eficiencia computacional entre ambas metodologías se encuentra en la sección “Resultados”.

1. ProbSecsCrup:

El pseudocódigo que ilustra el funcionamiento de esta función se encuentra en el Anexo C. Esta función cumple un rol similar al de *CrupierProb* en el código citado, basándose en información que puede ser pre calculada a partir de *CrupierProb*. Primero, se considera una baraja de ronda cualquiera “q”, a partir de la cual se debe ejecutar *CrupierProb* para los valores $w_c = (C = 0, z = 0, v = 0, q)$. Haciendo unas pequeñas modificaciones a *CrupierProb*, se rescatan todas las instancias finales que se alcanzan desde cada una de las 10 cartas iniciales que puede recibir el crupier, guardando además la cantidad de veces que se alcanza cada una (la multiplicidad de la instancia final). Notando que la diferencia de la baraja “q” con la baraja de las instancias finales corresponde a la secuencia de cartas que compone la mano del crupier, y que esta es independiente de la baraja “q” utilizada, luego es posible enumerar todos los casos que pueden ocurrir en una matriz.

Esta matriz se le llama *Matriz_Total*, y contiene en cada fila los valores ci , cf , sec_crup ,

mult. Estos valores son: el puntaje inicial del crupier con su primera carta (*ci*), el puntaje final que alcanza en la ronda (*cf*), la secuencia de cartas que componen la mano del crupier al final de la ronda incluyendo “*ci*” (*sec_crup*), y la multiplicidad de dicha secuencia de cartas excluyendo “*ci*” (*mult*). Esta matriz contiene alrededor de 4500 filas, y contiene la información necesaria para calcular CrupierProb (para cualquier valor de “C”, incluyendo C=0) a partir de una baraja “q” cualquiera.

La función *ProbSecsCrup* trabaja con esta matriz. Primero, la función considera si ya fue repartida la carta inicial del crupier (C distinto de 0), o si aún no se reparte (C=0). En caso de que C es distinto de 0, se actualiza la *sec_crup* de cada fila, extrayendo la carta asociada a *ci* de la secuencia. Luego, se procede a calcular la probabilidad de que ocurra *sec_crup* a partir de una baraja q, para cada fila, y se multiplica esta probabilidad por la multiplicidad de la secuencia, obteniendo la probabilidad real de que ocurra *sec_crup*. Esta probabilidad, *P_s*, se reemplaza por *mult* en la matriz, y se retorna una matriz análoga a *Matriz_Total*, llamada *Matriz_Probs*. Esta matriz contiene todas las secuencias de cartas que puede recibir el crupier a partir de un puntaje inicial C, con su respectiva probabilidad de ocurrencia partir de una baraja q, y es la información necesaria para deducir las probabilidades de los puntajes finales *cf* que puede obtener el crupier.

2. GanarPerder:

Esta función se encuentra ilustrada en el Anexo D, y se ha incluido par mayor claridad, a pesar de que no es particularmente relevante en la reformulación. La función recibe los puntajes e información asociada a la mano del jugador y del crupier, y a partir de *Matriz_Probs*, que contiene la probabilidad de cada posible secuencia del crupier, calcula la probabilidad de ganar y perder para el jugador si se planta. Primero, se filtra *Matriz_Probs* en caso de que la carta del crupier es conocida (C distinto de 0), y se comparan las variables J^1 , x^1 e y^1 con el puntaje final *cf* del crupier en cada fila, determinando si es una instancia ganadora (pago=1) o perdedora (pago=-1). Finalmente, se suman todas las probabilidades *P_s* asociadas a pago= 1 en *gpS[0]*, y en *gpS[1]* si pago= -1.

3. Bayes:

Esta función se encuentra ilustrada en el Anexo E y es fundamental para hacer más eficiente el código. La idea consiste en que la probabilidad de que ocurra una secuencia del crupier a partir de una baraja de ronda dada se puede calcular en términos de la probabilidad de que ocurra la misma secuencia frente a otra baraja de ronda distinta. La relación entre ambas probabilidades está dada por el teorema de bayes, y en particular resulta útil para actualizar la *Matriz_Probs* del crupier frente a una nueva baraja, sin tener que recalcular la probabilidad de cada secuencia. La función recibe una *Matriz_Probs* calculada para una determinada baraja, y deduce los valores de

Matriz_Probs para la misma baraja, pero habiéndole extraído una carta previamente. La función opera diferente para el caso $C = 0$ e $y^1 = 1$, pues justo en ese momento la carta extraída de la baraja va a la mano del crupier, por lo que el teorema se aplica levemente diferente.

4. **JugadorDecideRapido:**

JugadorDecide ha sido reformulada a partir de las funciones expuestas, de acuerdo con lo ilustrado en el pseudocódigo presentado en el Anexo F. En la formulación del trabajo referenciado, en cada llamado de *JugadorDecide* se calcula *CrupierProb*, a partir del cual se deduce el valor de gpS mediante la función que llamamos “Auxiliar”. La lógica ahora es similar. En el primer llamado de la función, en vez de calcularse *CrupierProb*, se inicializa *Matriz_Probs* a través de *ProbSecsCrup*(C, q), y se deduce gpS utilizando la función *GanarPerder*. Luego, se comienza a hacer las transiciones recursivas a nuevas instancias “sacando cartas” de la baraja q , de la misma manera que en *JugadorDecide*. Sin embargo, hay una diferencia fundamental. Mientras que *JugadorDecide* solo extrae una carta de la baraja y se llama a sí misma evaluada en una nueva instancia, *JugadorDecideRapido* realiza el mismo procedimiento, pero además genera la nueva *Matriz_Probs* que será utilizada en la nueva llamada recursiva, la cual es calculada a partir de la función *Bayes* y la matriz actual.

De esta manera, la nueva formulación deduce el vector gpS en cada nueva llamada recursiva de *JugadorDecideRapido* solo utilizando la función *Bayes* y la función *GanarPerder*, que son funciones de muy bajo costo computacional, en particular mucho más bajo que el costo de computar *CrupierProb*. Combinando esta metodología con paralelización, vectorización de algunos procedimientos y diccionarios para guardar y reutilizar información ya calculada en otras llamadas de la función, se logra finalmente la versión optimizada de *JugadorDecide*, que además es válida para cualquier instancia de la ronda.

5. **JugadorAbreExacta:**

Finalmente, se codificó *JugadorAbreExacta*, cuyo pseudocódigo ilustrativo se encuentra en el Anexo G. Esta función calcula la probabilidad de ganar o perder para el jugador si “Abre” su mano inicial de manera exacta, completando el análisis de la política de juego para todas las decisiones. Sin embargo, la función utiliza en cada una de las 2 manos abiertas, que se juegan secuencialmente, la política de juego optimizada para una mano individual a través de *JugadorDecideRapido*. Es decir, es una política sub-óptima, pues como se explicó, la cantidad de posibilidades jugando 2 manos secuencialmente crece de manera computacionalmente prohibitiva. El pseudocódigo ilustrativo de esta función considera nuevas versiones de *Bayes* y de *GanarPerder*, pero que son fundamentalmente análogas a las ya explicadas:

- *Bayes_aux*: Similar a *Bayes*, esta función calcula las probabilidades de las secuencias del crupier para una determinada baraja en términos de las probabilidades

contenidas en *Matriz_Probs*. La diferencia radica en que, si bien *Bayes* relaciona las probabilidades de las secuencias para 2 barajas separadas por una carta, *Bayes_aux* es mas general y considera barajas separadas por una secuencia de cartas. Además, no se utilizan las variables C e y^1 , pues este caso especial no ocurre en el contexto de *JugadorAbreExacta*.

- *GanarPerder_aux*: Mientras que *GanarPerder* retorna un vector gpS con 2 componentes (probabilidad de ganar y perder), *GanarPerder_aux* retorna un vector gpA con 4 componentes: [probabilidad de ganar ambas manos, probabilidad de perder ambas, probabilidad de ganar una mano y empatar la otra, probabilidad de perder una mano y empatar la otra]. Estos 4 casos capturan todas las posibilidades considerando ambas manos del crupier, ignorando los casos en que el jugador “empata” con el crupier, pues no son necesarios para la esperanza.

6. Segunda Parte: Políticas de Apuestas

6.1. Sesión de Apuestas

Hemos logrado definir y calcular en tiempo real de la FMP del retorno por ronda, solo sabiendo el estado de la baraja antes del comienzo de la ronda. *Se destaca la importancia de esta función*, pues determina la forma matemática explícita en que distribuye el retorno en cada ronda antes de realizar una apuesta. Este es el escenario de análisis financiero teórico ideal, pues permite modelar los retornos por periodo de manera exacta a partir de información disponible.

Sin embargo, dado que en el Blackjack es común que el jugador participe de una serie de rondas de apuestas, conformando una *sesión de apuestas*, la política de juego óptima no es suficiente. En esta sección se abordará la problemática que enfrenta el jugador antes del comienzo de cada ronda, frente a una baraja de origen cualquiera, que consiste en determinar *qué porcentaje de su patrimonio apostar*. A la estrategia que determina cuando y cuanto apostar ronda tras ronda en una sesión de apuestas le llamaremos *política de apuestas*, y en particular nos interesa modelar y estudiar las vías para encontrar la política de apuestas óptima, que resulta ser un problema de naturaleza más compleja que de la política de juego.

6.1.1. Secuencia de Barajas

Partiendo del supuesto que el jugador conoce y aplica la política de juego óptima en cada momento, el primer paso para modelar una sesión de apuestas consiste en comprender como evoluciona la baraja de origen a la que se enfrenta el jugador ronda tras ronda. Notando que el retorno óptimo por ronda $X_{\theta^*}^R$ depende de la baraja de origen R , se hace evidente que existe correlación entre retornos asociados a distintos rondas, pues desde una cierta baraja de origen R solo es posible transicionar a un determinado subconjunto de $\Omega_{c,m}$ en cada ronda, limitado por la cantidad máxima de cartas que se pueden jugar en la ronda actual. Se define entonces \mathcal{Y}_{θ^*} , como una secuencia de barajas de origen aleatorias jugando bajo la política de juego óptima θ^* , a partir de una baraja de origen completa, como sigue:

Secuencia de Barajas de Origen:

$$\mathcal{Y}_{\theta^*} : \{B_n\}_{0 \leq n}$$

1. Tal que B_n es la baraja de origen de la n -ésima ronda, cuya FMP se define como sigue:

$$C_{\theta^*}(b_{n+1}, b_n) = \text{Prob}(B_{n+1} = b_{n+1} | B_n = b_n, \theta^*)$$

Recordando la función de probabilidad absoluta D_{θ^*} :

$$C_{\theta^*}(b_{n+1}, b_n) = \begin{cases} \sum_{w' \in \mathcal{W}^{F, b_n}} D_{\theta^*}(w', w_0) \cdot 1_{\{q' = b_{n+1}\}} & , \text{ si } b_{n+1} \neq B_0 \\ \sum_{w' \in \mathcal{W}^{F, b_n}} D_{\theta^*}(w', w_0) \cdot 1_{\{q' \notin \Omega_{c, m}\}} & , \text{ si } b_{n+1} = B_0 \end{cases}$$

Con:

- $B_0 = [4m, 4m, 4m, 4m, 4m, 4m, 4m, 4m, 4m, 16m]$: Baraja inicial del juego, con “m” mazos.
- $w_0 = (0, 0, 0, 0, 0, 0, 0, 0, 0, b_n)$: Instancia de origen de la enésima ronda.
- $w' = (\dots, q') \in \mathcal{W}^{F, b_n}$: Instancia final de la enésima ronda.

6.1.2. Ecuación para el Patrimonio

Ahora, procedemos a modelar la evolución de la situación de un apostador de Blackjack a través de una secuencia de barajas de origen. Sea la secuencia de barajas de origen $\mathcal{Y}_{\theta^*} = \{B_0, B_1, B_2, \dots\}$ bajo la política de juego óptima θ^* . Sea P_n el patrimonio del jugador previo al comienzo de la ronda n . Sea π_n la apuesta realizada por el jugador previo al comienzo de la ronda n , como una fracción del patrimonio P_n . A partir de estas variables y el retorno por ronda $X_{\theta^*}^{B_n}$ para la baraja de origen B_n , se deduce la ecuación recursiva que define el patrimonio del jugador a lo largo de una *sesión de apuestas*, a partir de un patrimonio inicial normalizado $P_0 = 1$:

Ecuación para el Patrimonio:

$$P_{n+1} = (1 + \pi_n \cdot X_{\theta^*}^{B_n}) \cdot P_n$$

Con:

$$B_n \in \mathcal{Y}_{\theta^*}$$

$$P_0 = 1$$

$$\pi_n \in [0, 0,5], \quad \forall n \geq 0.$$

Donde la apuesta fraccionaria realizada por el jugador π_n previa a cada ronda se restringe al intervalo $[0, 0.5]$, de tal manera que el jugador siempre tenga la chance de doblar o abrir su apuesta inicial, y por lo tanto sea consistente el uso de la variable $X_{\theta^*}^{B_n}$ en el modelo, que asume que el jugador siempre puede realizar estas acciones en cada ronda.

6.1.3. Política de Apuestas

Como se mostró, la composición de la baraja de origen B_n , las reglas del juego y la política de juego θ^* , determinan de manera exacta la FMP del retorno de cada ronda. Sin embargo, antes de recibir una mano de cartas y jugar la ronda, el jugador debe enfrentarse a una decisión previa mucho más importante: cuánto apostar. Es decir, *el jugador debe determinar*

qué porcentaje del patrimonio se apuesta en cada ronda, de acuerdo con algún criterio u objetivo específico. Sin embargo, como veremos, esto no es un problema sencillo.

En la ecuación que describe el patrimonio del jugador P_n , hemos introducido de manera general la función o *política de apuestas* π_n , que entrega la apuesta realizada previa a cada ronda como una fracción del patrimonio actual del jugador. Para establecer la estructura general del argumento de esta función, es necesario identificar las variables que determinan la transición de P_n a P_{n+1} , de tal manera que son relevantes para la evolución del patrimonio, que es la variable de interés para el apostador. Observando la ecuación para el patrimonio, notamos que la baraja de origen de la ronda $B_n = b_n$ y el patrimonio actual $P_n = p_n$ son suficientes para conocer la función de transición de P_n a P_{n+1} , según la apuesta π_n que se haga. Sin embargo, dado que el jugador puede estar interesado en el patrimonio P_H de una sesión de apuestas con una cantidad específica de rondas $n = H$, se incorpora adicionalmente la ronda actual n como variable para la política.

Por lo tanto, las variables $B_n = b_n$ y $P_n = p_n$ determinan completamente la función de transición del patrimonio en cada ronda, mientras que la ronda actual n toma relevancia específicamente respecto a la cantidad de rondas que componen la sesión de apuestas. Estas 3 variables corresponden a las *variables de estado*, y definen el argumento de la política de apuestas.

Formalizamos entonces la *política de apuestas* π , como aquella función que retorna una apuesta porcentual, a partir de una baraja de origen b , el valor del patrimonio actual p , y la cantidad de rondas n jugadas desde el comienzo de una sesión de apuestas.

Política de Apuesta:

$$\begin{aligned} \pi : \Omega_{c,m} \times \mathbb{R}_+ \times \mathbb{Z}_+ &\rightarrow [0, 0,5] \\ (b, p, n) &\rightarrow \pi(b, p, n) \end{aligned}$$

Finalmente, la ecuación para el patrimonio queda definida la siguiente manera:

Patrimonio bajo Política:

$$P_{n+1} = (1 + \pi(B_n, P_n, n) \cdot X_{\theta^*}^{B_n}) \cdot P_n$$

Con:

$$B_n \in \mathcal{Y}_{\theta^*}$$

$$P_0 = 1$$

6.2. Modelamiento como Proceso de Decisión de Markov

El problema de determinar la política de apuestas óptima para el jugador se puede modelar como un *Proceso de Decisión de Markov* (PDM). En este tipo de procesos, un agente interactúa con el entorno tomando decisiones secuenciales por etapa, de acuerdo con una *política*, y recibe una recompensa asociada con la *acción* ejecutada y el *estado* resultante. Según la política establecida para la determinación de la acción a partir del estado actual, la *probabilidad de transición* entre estados se ve afectada, lo cual repercute en la evolución estocástica de la situación del agente.

Una vez el agente tiene una política establecida, luego la evolución de su situación se puede modelar como una Cadena de Markov, lo que significa que la distribución de probabilidad del estado para el próximo periodo depende únicamente del estado en el periodo actual y la acción tomada, y no de los estados y acciones previos o futuros. El problema se reduce a encontrar una política óptima, que solucione el *Problema de Optimización* definido por el agente a través de un criterio de optimalidad establecido por una “función de recompensa”. Veamos cómo aplicar estos conceptos a nuestro caso de estudio.

6.2.1. Estados

Como se mostró en la definición de la política de apuestas, el *estado* $s_n = (B_n, P_n, n)$ del apostador en la ronda n de una sesión de apuestas se define completamente por la baraja de origen B_n , el patrimonio actual P_n , y el número de la ronda, n . Se define así el *espacio de estados* que puede tomar la situación del apostador, que coincide con el dominio de la política de apuestas:

Espacio de Estados:

$$E = \Omega_{c,m} \times \mathbb{R}_+ \times \mathbb{Z}_+$$

6.2.2. Función de Transición

Ahora, según la acción o apuesta fraccionaria “ a ” que realiza el apostador en un determinado estado, luego su probabilidad de transicionar a otros estados se ve afectada. Es decir, la decisión de cuánto apostar determina el espacio de estados a los que puede transicionar el jugador, junto con las respectivas probabilidades de transición. Se define entonces la *función de transición* $T_a(s^*, s)$, que calcula la probabilidad de transicionar desde el estado s al estado s^* luego de jugarse una ronda, habiendo apostado una fracción “ a ” del patrimonio al inicio de la ronda:

Función de Transición:

$$T : E \times E \rightarrow [0, 1]$$

$$T_a(s^*, s) = \text{PROB}(s_{n+1} = s^* | s_n = s, \pi_n = a)$$

Utilizando las funciones definidas en la sección “Política de Juego”, se deduce la forma explícita para la función de transición:

Si $b^* \neq B_0$:

$$T_a(s^*, s) = \sum_{w' \in \mathcal{W}^{F,b}} D_{\theta^*}(w', w_0) \cdot \left(\sum_{y \in \mathcal{P}_{\text{agosBJ}}} f(y, w') \cdot 1_{\{p^* = (1+a \cdot y)p\}} \right) \cdot 1_{\{q' = b^*\}}$$

Si $b^* = B_0$:

$$T_a(s^*, s) = \sum_{w' \in \mathcal{W}^{F,b}} D_{\theta^*}(w', w_0) \cdot \left(\sum_{y \in \mathcal{P}_{\text{agosBJ}}} f(y, w') \cdot 1_{\{p^* = (1+a \cdot y)p\}} \right) \cdot 1_{\{q' \in \Omega_{c,m}\}}$$

Con:

$s = (b, p, n) \in E$.

$s^* = (b^*, p^*, n+1) \in E$.

$w_0 = (0, 0, 0, 0, 0, 0, 0, 0, 0, b)$: Instancia de origen asociada al estado s .

$w' = (\dots, q') \in \mathcal{W}^{F,b}$: Instancia final a partir de la instancia de origen w_0 .

6.3. Problema de Optimización y Función de Recompensa

Si bien se ha definido explícitamente la función de transición durante una sesión de apuestas, para que un Proceso de Decisión de Markov esté completamente definido se ha de introducir una noción de “premio” o recompensa *inmediata* asociada a la transición entre estados, que permita al apostador elaborar un criterio para jerarquizar ciertas estrategias o *políticas* sobre otras. Este premio se formaliza a través de una *función de recompensa* $R_a(s^*, s)$, cuya forma está directamente relacionada con el *Problema de Optimización*. En esta sección se detalla primero la forma matemática que toma el problema de optimización en el contexto de un PDM, para luego discutir su relación con la función de recompensa.

6.3.1. Problema de Optimización

Sea $R_{a_n}(s_{n+1}, s_n)$ la recompensa inmediata que experimenta el apostador al transicionar desde el estado s_n al estado s_{n+1} , tras haber realizado una apuesta fraccionaria de $a_n = \pi(s_n)$ en la ronda n (esta función se discute detalladamente en la próxima sección, pero podemos imaginar que su valor depende de p_{n+1} y p_n). Se introduce entonces la función de *Recompensa Parcial Esperada* $U_\pi^{(s_k, H)}$, que calcula la esperanza de la recompensa acumulada parcial que

obtiene el apostador desde la ronda k hasta la ronda $H - 1$, bajo una política de apuestas π . Esta es una función auxiliar que servirá para definir la *Recompensa Total Esperada* del jugador, y se define como sigue:

Recompensa Parcial Esperada:

$$U_{\pi}^{(s_k, H)} = \mathbb{E} [A_{\pi}^{(s_k, H)}] = \mathbb{E} \left[\sum_{n=k}^{H-1} R_{\pi(s_n)}(s_{n+1}, s_n) \right]$$

Con:

$s_k = (B_k, P_k, k) \in E$: Estado del apostador en la ronda k .

$H \in \mathbb{N}$: Cantidad de rondas que planea jugar el apostador.

$A_{\pi}^{(s_k, H)}$: Recompensa acumulada desde la ronda k hasta la ronda $H - 1$.

El valor de esta función puede ser calculado recursivamente como sigue:

$$U_{\pi}^{(s_k, H)} = \begin{cases} \sum_{s^* \in E} T_{\pi(s_k)}(s^*, s_k) \left[R_{\pi(s_k)}(s^*, s_k) + U_{\pi}^{(s^*, H)} \right], & \text{si } k < H \\ 0 & \text{si } k \geq H \end{cases}$$

La función $U_{\pi}^{(s_k, H)}$ calcula recursivamente la recompensa acumulada durante las $(H - k)$ rondas comprendidas entre un estado intermedio s_k y el estado terminal s_H . Sin embargo, un caso particular de esta función se tiene cuando se considera la recompensa acumulada desde la ronda inicial $n = 0$ de la sesión de apuestas, pues en este caso la función calcula la *recompensa acumulada total* durante toda la sesión, que es la que finalmente resulta relevante para el apostador. Se introduce entonces la función *Recompensa Total Esperada* $U_{\pi}^{(H)}$, que calcula la esperanza de la *recompensa acumulada total* $A_{\pi}^{(H)}$ que obtiene el apostador desde el inicio de la sesión de apuestas hasta la ronda H :

Recompensa Total Esperada:

$$U_{\pi}^{(H)} \leftrightarrow U_{\pi}^{(s_0, H)}$$

$$U_{\pi}^{(H)} = \mathbb{E} [A_{\pi}^{(H)}] = \mathbb{E} \left[\sum_{n=0}^{H-1} R_{\pi(s_n)}(s_{n+1}, s_n) \right]$$

Con:

$s_0 = (B_0, P_0, 0) \in E$: Estado inicial de la sesión de apuestas, con baraja de origen y patrimonio establecidos.

* $A_{\pi}^{(H)}$: *Recompensa acumulada total durante H rondas de apuestas. No se ha incorporado*

el factor de descuento para recompensas futuras, debido a que no es coherente en el contexto de una sesión de apuestas, cuya duración puede ser de horas o a lo más un día. Sin embargo, si se decidiera establecer estrategias de apuestas para el largo plazo, como años, sí podría resultar coherente incorporar un factor de descuento temporal para las recompensas.

Finalmente, el problema de optimización para la política de apuestas consiste justamente en la maximización de la recompensa total esperada del jugador durante una sesión de H rondas de apuestas:

Problema de Optimización:

$$\pi_H^*(s) = \operatorname{argmax}_\pi \mathbb{E}[A_\pi^{(H)}]$$

Encontrar la política π_H^* significa maximizar la esperanza de la recompensa acumulada total del jugador para una determinada cantidad de rondas de apuestas H .

Resulta importante destacar que si bien la recompensa total $A_\pi^{(H)}$ y la función de recompensa inmediata $R_a(s^*, s)$ se definen mutuamente, la definición del problema de optimización se ha de formular a partir de $A_\pi^{(H)}$, pues es esta función la que captura directamente aquello que el jugador considera valioso. A continuación, se plantea cómo definir adecuadamente la función $A_\pi^{(H)}$, de tal manera que pueda ser interpretada económicamente, para luego deducir la función de recompensa inmediata correspondiente.

6.3.2. Función de Recompensa

Si bien para el caso de la política de juego la esperanza del retorno de la ronda es suficiente como criterio de optimalidad para la política, el caso de la política de apuestas es más complejo, pues no existe un criterio universal que satisfaga a todos los apostadores, y por lo tanto no existe un único problema de optimización para la política.

Como se mostró en la sección anterior, el problema de optimización se plantea en términos de la esperanza de la recompensa total de la sesión de apuestas, $A_\pi^{(H)}$. Sin embargo, notando que la satisfacción del apostador al final de una sesión de apuestas depende únicamente del valor de su patrimonio al final de la sesión, sintiéndose más satisfecho cuando este es mayor, por ejemplo, luego es lógico asumir que *la recompensa total debe estar definida en términos del patrimonio final P_H del jugador al final de la sesión de apuestas.*

Sin embargo, distintos apostadores pueden valorar de manera diferente los mismos resultados, según sus propias inclinaciones y apreciación de lo que es más conveniente. Por esta razón, se vuelve necesario modelar la valoración personal del jugador, en particular a través de una función, que determine el valor o utilidad que el apostador le asigna al patrimonio final de una sesión de apuestas. Se introduce entonces la *función de utilidad u* , que recibe

como argumento el patrimonio y retorna el valor que se asigna a dicho patrimonio:

Función de Utilidad:

La función de utilidad r se define como:

$$r : \mathbb{R}_+ \rightarrow \mathbb{R}$$
$$p \rightarrow r(p)$$

Si bien la función de utilidad es absolutamente general en principio, esta debe ser consistente con el principio de jugador racional que busca obtener ganancias, valorando siempre más el patrimonio cuando es mayor a 1 (ganancias) que cuando es menor a 1 (pérdidas), por ejemplo. Luego, se propone una forma específica para la función de utilidad, con un parámetro libre γ , que *identificamos con el nivel de aversión al riesgo del apostador*:

Utilidad del Apostador:

$$r_\gamma : \mathbb{R}_+ \rightarrow \mathbb{R}$$
$$p = \begin{cases} r_\gamma(p) = \frac{p^\gamma}{\gamma} & , \gamma \neq 0 \\ r_\gamma(p) = \ln(p) & , \gamma = 0 \end{cases}$$

Con $\gamma \in \mathbb{R}$, donde el sentido de la función se visualiza a través de su derivada, que se interpreta como la *utilidad marginal* de obtener más patrimonio a partir de un patrimonio p :

Utilidad Marginal del Apostador:

La utilidad marginal del apostador $m_\gamma(p)$ se define como la derivada de la función de utilidad respecto al patrimonio p :

$$m_\gamma(p) = \frac{dr_\gamma(p)}{dp} = p^{\gamma-1}$$

Aspectos relevantes de la función de utilidad $r_\gamma(p)$ y la función de utilidad marginal $m_\gamma(p)$:

1. La utilidad marginal del apostador $m_\gamma(p)$ es *positiva y continua para todo nivel de riesgo* γ . Esto significa que para todo nivel de riesgo siempre se tiene una utilidad creciente del patrimonio (siempre se valora más un patrimonio mayor que otro menor), lo cual es consistente con el principio de apostador racional.
2. La derivada de la utilidad marginal $m_\gamma(p)$ *respecto del riesgo* γ es $\ln(p) \cdot p^{\gamma-1}$, la cual es negativa para $p < 1$ y positiva para $p > 1$. Esto quiere decir que a mayor riesgo aumenta la utilidad marginal del apostador cuando va ganando, y disminuye cuando va perdiendo. Esta propiedad matemática es consistente con la interpretación de γ como

el nivel de riesgo del apostador.

3. $\gamma = 0$: Este nivel de riesgo valora el patrimonio de manera logarítmica $r_0(p) = \ln(p)$, y su utilidad marginal es inversamente proporcional al monto del patrimonio $m_0(p) = p^{-1}$. Se considera un nivel de riesgo *neutro*. Cuando $\gamma > 0$ diremos que es una valorización de riesgo *alto*, y cuando $\gamma < 0$ se entenderá por una valorización de riesgo *bajo*.
4. $\gamma = 1$: Este nivel de riesgo valora el patrimonio de manera absoluta $r_1(p) = p$, y su utilidad marginal de obtener más patrimonio es constante e igual a uno $m_1(p) = 1$.

Ahora estamos en condiciones de definir explícitamente la función de recompensa acumulada total $A_\pi^{(H)}$ en términos de la utilidad del patrimonio final $r_\gamma(p)$ que define al apostador. Definimos entonces la recompensa para el apostador $A_\pi^{(H,\gamma)}$, como la utilidad del patrimonio final $r_\gamma(P_H)$ de la sesión de apuestas, que es igual a la recompensa total para la misma sesión:

Recompensa para el Apostador:

$$A_\pi^{(H)} = A_\pi^{(H,\gamma)} = r_\gamma(P_H)$$

Finalmente, el problema de optimización para la política de apuestas queda completamente definido, incorporando una cierta cantidad de rondas H y un nivel de riesgo γ asociado al apostador. A este problema le llamaremos $O_{\gamma,H}$ y resulta como sigue:

$$\pi_{\gamma,H}^*(s) = \operatorname{argmax}_\pi \mathbb{E}[r_\gamma(P_H)]$$

Si bien la función r_γ permite definir un problema de optimización para la política, capturando la inclinación personal del apostador, aún se debe establecer la relación entre dicha función, que se corresponde con la recompensa total, con las recompensas inmediatas que obtiene el jugador a lo largo de la sesión. En particular, debemos encontrar la función $R_a(s^*, s)$ que cumpla la siguiente relación:

Relación Recompensa Total - Recompensa Inmediata:

$$r_\gamma(P_H) = \sum_{n=0}^{H-1} R_{\pi(s_n)}(s_{n+1}, s_n) = A_\pi^{(H,\gamma)}$$

Dado que el parámetro γ implica un valor diferente de r_γ , la función de recompensa inmediata que cumpla la relación también debe incorporarlo. Llamamos entonces como $R_\gamma^{(\alpha)}(s^*, s)$ a la función de recompensa inmediata asociada a la recompensa total $A_\pi^{(H,r)}$, y observamos que la función r_γ se puede reescribir como:

$$r_\gamma(P_H) = r_\gamma(P_0) + \sum_{k=0}^{H-1} [r_\gamma(P_{k+1}) - r_\gamma(P_k)]$$

Luego la relación establecida para la recompensa total e inmediata se puede reformular como sigue:

$$r_\gamma(P_0) + \sum_{k=0}^{H-1} [r_\gamma(P_{k+1}) - r_\gamma(P_k)] = \sum_{n=0}^{H-1} R_{\pi(s_n)}(s_{n+1}, s_n)$$

Con $P_0 = 1$.

Finalmente, el problema se puede dividir en 2 casos:

1. Caso $\gamma = 0$:

$$\sum_{k=0}^{H-1} [\ln(P_{k+1}) - \ln(P_k)] = \sum_{n=0}^{H-1} R_{\pi(s_n)}^{(0)}(s_{n+1}, s_n)$$

Haciendo la equivalencia entre cada término de ambas sumas, y recordando que $s_n = (B_n, P_n, n)$, se deduce que la función $R_a^{(0)}$ que cumple la relación es como sigue:

$$R_a^{(0)}(s^*, s) = \ln(p^*) - \ln(p)$$

2. Caso $\gamma \neq 0$:

$$1 + \sum_{k=0}^{H-1} \left[\frac{(P_{k+1})^\gamma - (P_k)^\gamma}{\gamma} \right] = \sum_{n=0}^{H-1} R_{\pi(s_n)}^{(\gamma)}(s_{n+1}, s_n)$$

Ignorando el “1” al lado izquierdo de la ecuación (dado que sumar o multiplicar por una constante positiva la recompensa total lleva a la misma solución del problema de optimización por las propiedades de la esperanza) y considerando solo el cambio de signo asociado al denominador “ γ ” (normalizando su valor, por las propiedades de la esperanza), se deduce la forma para la función de recompensa:

$$R_a^{(\gamma)}(s^*, s) = \begin{cases} (p^*)^\gamma - p^\gamma, & \text{si } \gamma > 0 \\ p^\gamma - (p^*)^\gamma, & \text{si } \gamma < 0 \end{cases}$$

Se concluye entonces con la forma explícita de la función de recompensa inmediata $R_{(\alpha)}^\gamma$ asociada a la valorización del apostador r_γ :

Función de Recompensa del Apostador:

$$R_a^{(\gamma)}(s^*, s) = \begin{cases} \ln(p^*) - \ln(p), & \text{si } \gamma = 0 \\ (p^*)^\gamma - p^\gamma, & \text{si } \gamma > 0 \\ p^\gamma - (p^*)^\gamma, & \text{si } \gamma < 0 \end{cases}$$

Por lo tanto, el problema de encontrar la política de apuestas óptima queda completamente definido como un Proceso de Decisión de Markov, para el cual se ha explicitado la función de transición, un problema de optimización concreto coherente con los principios económicos, y la forma explícita de la función de recompensa.

***NOTA:** Como se mostró, si bien se tiene que $r_\gamma(p) = p^\gamma \cdot \gamma^{-1}$ para $\gamma \neq 0$, el problema de optimización consiste en maximizar $\mathbb{E}[(P_H)^\gamma \cdot \gamma^{-1}]$, que es equivalente con maximizar $\mathbb{E}[(P_H)^\gamma]$ para $\gamma > 0$ y maximizar $\mathbb{E}[-(P_H)^\gamma]$ para $\gamma < 0$. La incorporación de γ^{-1} en la definición fue con la motivación de mantener la continuidad de la función de valorización marginal respecto al riesgo para mayor claridad, sin embargo, en la práctica resulta más práctico usar simplemente $r_\gamma(p) \pm p^\gamma$ (según el signo de γ) como función de valorización, manteniendo el problema de optimización equivalente.

6.4. Solución del Problema

A través de las secciones previas, se ha logrado definir completamente un problema de optimización para la política de apuestas como un Proceso de Decisión de Markov. Sin embargo, contrario al caso de la política de juego, el problema de optimización de la política de apuestas no se puede resolver directamente mediante programación dinámica, y por lo tanto no es posible hallar la solución general del problema de manera exacta. Esto se debe a que el espacio de estados que puede enfrentar la política de apuestas y el espacio de acciones que puede ejecutar son continuos y por lo tanto infinitos, mientras que la cantidad de estados (instancias) que enfrenta la política de juego y la cantidad de acciones que puede ejecutar son finitos y manejables. Esta clase de PDM son más complejos, y son abordados generalmente a través de técnicas de programación dinámica en conjunto con Redes Neuronales por Refuerzo. Dado que esta metodología escapa a los alcances del presente trabajo, la solución general del problema no será abordada en esta investigación.

Sin embargo, existe una versión simplificada del problema que sí admite solución teórica, correspondiente al caso en que la política depende únicamente de la baraja de origen B_n de la ronda actual, ignorando el patrimonio actual P_n y el número de rondas jugadas n . A esta forma simplificada de la política la llamaremos *política parcial*, y desarrollaremos el problema de optimización $O_{\gamma,H}$ a nivel teórico, considerando el caso en que la sesión de apuestas se juega eternamente $H \rightarrow \infty$, culminando con una ecuación fácilmente resoluble (mediante métodos computacionales) para hallar la solución exacta.

6.4.1. Política Dependiente de la Baraja

Como se mostró, la forma completa de la política incorpora como variables tanto el estado de la baraja de origen de la ronda, el patrimonio actual y el número de rondas jugadas desde el comienzo de la sesión de apuestas. Si bien la optimización de la política completa no es posible mediante métodos analíticos, podemos imaginar que el apostador solo considera un subconjunto de las variables de estado para decidir su apuesta. Se define entonces g como la política solo dependiente de la baraja de origen b de una ronda, y que llamaremos *política parcial*, como sigue:

Política Parcial:

$$g : \Omega_{cm} \rightarrow [0, 0,5]$$

$$b \rightarrow g(b)$$

Luego, haciendo la simplificación de notación $g(b) = g^b$, la ecuación para el patrimonio resulta:

Ecuación para el Patrimonio bajo Política Parcial:

$$P_{n+1} = (1 + g^{B_n} \cdot X_{\theta^*}^{B_n}) \cdot P_n$$

Con:

$$B_n \in \mathcal{Y}_{\theta^*}.$$

$$P_0 = 1.$$

A continuación, procederemos a tratar matemáticamente la ecuación para el patrimonio recién expuesta, con la intención de deducir la forma algebraica de la esperanza de la recompensa total $\mathbb{E}[r_\gamma(P_{H \rightarrow \infty})]$, y poder así optimizar la política. Notando que la ecuación es un producto iterado, tomamos el logaritmo natural de P_{n+1} para dejar la ecuación como una suma en vez de producto:

$$\ln(P_{n+1}) = \sum_{k=0}^n \ln(1 + g^{B_k} \cdot X_{\theta^*}^{B_k})$$

Hasta ahora, la ecuación sigue siendo irresoluble analíticamente, principalmente por la correlación que existe entre cada término de la suma, debido a la correlación de las barajas de origen B_k . Sin embargo, si nos preguntamos sobre el comportamiento de $\ln(P_{n+1})$ cuando $n \rightarrow \infty$, encontraremos que la ecuación tiende a la suma de variables aleatorias independientes, y por lo tanto podemos aplicar teoremas convenientes.

Notemos entonces que las barajas B_k solo pueden tomar un conjunto finito de valores b en $\Omega_{c,m}$, dado que $M = \text{card}(\Omega_{c,m})$ es una cantidad finita, como se mostró en secciones anteriores.

Esto implica que si jugamos una cantidad infinita de rondas $n \rightarrow \infty$, entonces cada baraja de origen $b \in \Omega_{c,m}$ que pueden tomar como valor las barajas B_n en \mathcal{Y}_{θ^*} , necesariamente se repetirá una cantidad indefinida de veces a medida que se juegan más rondas. A nivel técnico esto se debe a que la cadena \mathcal{Y}_{θ^*} es *recurrente positiva*, y por lo tanto todos los estados b^j que puede tomar la variable B_n poseen un tiempo medio de recurrencia igual a u_j , el cual permite calcular la frecuencia $p_{b^j} = \frac{1}{u_j}$ con que ocurre la baraja b^j dentro de la secuencia \mathcal{Y}_{θ^*} en un plazo infinito.

El valor de p_{b^j} se podría deducir matemáticamente a partir de la definición de \mathcal{Y}_{θ^*} , pero debido a que dicho cálculo es engorroso y en la práctica no tiene utilidad, nos limitaremos a definir su valor formalmente. Dicho esto, definimos la frecuencia de una baraja de origen $v \in \Omega_{c,m}$ en la secuencia \mathcal{Y}_{θ^*} , cuando se juegan infinitas barajas, a través de la variable V , que representa una baraja promedio de la secuencia \mathcal{Y}_{θ^*} :

Baraja Promedio:

$$V : \Omega_{c,m} \rightarrow [0, 1]$$

$$p_v = \text{Prob}(V = v) = \lim_{z \rightarrow \infty} \frac{\sum_{n=0}^z \text{Prob}(B_n = v)}{z + 1}$$

Con:

$$B_n \in \mathcal{Y}_{\theta^*}$$

Asumiendo que se conoce el valor p_v para cada $v \in \Omega_{c,m}$, retomamos la ecuación que definimos para $\ln(P_{n+1})$, y deducimos el siguiente resultado, a partir del Teorema del Límite Central:

$$\lim_{n \rightarrow \infty} \sum_{k=0}^n \ln(1 + g^{B_k} \cdot X_{\theta^*}^{B_k}) = \lim_{n \rightarrow \infty} \sum_{v \in \Omega_{c,m}} p_v \cdot (n + 1) \cdot \ln(1 + g^v \cdot X_{\theta^*}^v)$$

Por lo tanto, se concluye el siguiente resultado para $\ln(P_{n+1})$ y P_{n+1} en el límite $n \rightarrow \infty$:

$$\lim_{n \rightarrow \infty} \ln(P_{n+1}) = \lim_{n \rightarrow \infty} \sum_{k=0}^n \ln(1 + g^{V_k} \cdot X_{\theta^*}^{V_k})$$

$$\lim_{n \rightarrow \infty} P_{n+1} = \lim_{n \rightarrow \infty} \prod_{k=0}^n (1 + g^{V_k} \cdot X_{\theta^*}^{V_k})$$

Con V_k variables aleatorias independientes e idénticamente distribuidas con distribución V .

Notamos que se tiene una suma de infinitas variables aleatorias independientes e idénticamente distribuidas con distribución V .

ticamente distribuidas $S_g^V = \ln(1 + g^V \cdot X_{\theta^*}^V)$. Por lo tanto, si encontramos la esperanza y desviación estándar de S_g^V , estaremos en condiciones de aplicar el Teorema del Límite Central, y finalmente deducir la distribución de $\ln(P_{n+1})$. Procedemos entonces a calcular la esperanza μ y desviación estándar σ de la variable aleatoria S_g^V , dada la política parcial g^b :

Esperanza:

$$\mu_g = \mathbb{E}(S_g^V) = \sum_{v \in \Omega_{c,m}} p_v \cdot \mathbb{E}[\ln(1 + g^v \cdot X_{\theta^*}^v)]$$

$$\mu_g = \mathbb{E}(S_g^V) = \sum_{v \in \Omega_{c,m}} p_v \cdot \left(\sum_{x \in \text{PagosBJ}} \text{Prob}(X_{\theta^*}^v = x) \cdot \ln(1 + g^v \cdot x) \right)$$

Desviación Estándar:

$$\sigma_g^2 = \text{Var}(S_g^V) = \sum_{v \in \Omega_{c,m}} p_v \cdot \mathbb{E} [(\ln(1 + g^v \cdot X_{\theta^*}^v) - \mu_g)^2]$$

$$\sigma_g^2 = \text{Var}(S_g^V) = \sum_{v \in \Omega_{c,m}} p_v \cdot \left(\sum_{x \in \text{PagosBJ}} \text{Prob}(X_{\theta^*}^v = x) (\ln(1 + g^v \cdot x) - \mu_g)^2 \right)$$

Ya conociendo la esperanza y desviación estándar de $S_g^V = \ln(1 + g^V \cdot X_{\theta^*}^V)$, concluimos por Teorema del Límite Central que la variable $\ln(P_{n+1})$ tiende a una distribución normal cuando $n \rightarrow \infty$:

Distribución para gamma = 0:

$$r_0(P_{n+1}) = \ln(P_{n+1}) \sim \text{Normal}(n \cdot \mu_g, n \cdot \sigma_g^2)$$

Tal que la Función de Distribución Acumulada (FDA) es como sigue, donde Φ se refiere a la FDA de una normal estándar:

$$\text{Prob}(\ln(P_{n+1}) \leq p) = \Phi \left(\frac{p - n \cdot \mu_g}{\sqrt{n} \cdot \sigma_g} \right)$$

Cuando $n \rightarrow \infty$.

Ahora, dado que estamos en una situación donde el logaritmo de P_{n+1} sigue una distri-

bución Normal, luego se tiene que el patrimonio P_{n+1} sigue una distribución LogNormal, y por propiedades de la distribución log normal finalmente se deduce la distribución que sigue P_{n+1}^γ para $\gamma \neq 0$:

Distribución para gamma $\neq 0$:

$$r_\gamma(P_{n+1}) = P_{n+1}^\gamma \sim \text{LogNormal}(n \cdot \gamma \cdot \mu_g, n \cdot \gamma^2 \cdot \sigma_g^2)$$

Tal que la Función de Distribución Acumulada (FDA) es como sigue, donde Φ se refiere a la FDA de una normal estándar:

$$\text{Prob}(P_{n+1}^\gamma \leq p) = \Phi \left(\frac{\ln(p) - n \cdot \gamma \cdot \mu_g}{\sqrt{n \cdot \gamma \cdot \sigma_g}} \right)$$

Cuando $n \rightarrow \infty$.

Con:

$\gamma \neq 0$.

**Un caso particular sucede cuando la política apuesta 0.5 para alguna baraja en específico $g(b) = 0.5$. Cuando esto ocurre, se tiene que $\mathbb{E}(S_g^V) = \mu_g \rightarrow -\infty$, pues $\ln(0) \rightarrow -\infty$. Esto significa que en una sesión de apuestas infinita se llega casi seguramente a la ruina si se determina apostar la mitad del patrimonio para alguna baraja en específico: $\lim_{n \rightarrow \infty} \text{Prob}(r_\gamma(P_n) = 0) \rightarrow 1$. Esto es consistente con los resultados para Cadenas de Markov absorbentes y el Teorema de la Ruina del jugador, y conduce a paradojas económicas, como se verá más adelante.*

6.4.2. Política Parcial Óptima

Hemos encontrado la distribución de probabilidad que sigue $r_\gamma(P_{H \rightarrow \infty})$. En este apartado procederemos a abordar el problema de optimización $O_{\gamma, H \rightarrow \infty}$, que se refiere a la maximización de la esperanza de la utilidad $\mathbb{E}[r_\gamma(P_{H \rightarrow \infty})]$ cuando la sesión de apuestas se juega por un tiempo indefinidamente largo. En particular, el problema de optimización se puede separar en casos como sigue:

$$O_{\gamma, H \rightarrow \infty} : \begin{cases} g_{\gamma, \infty}^*(b) = \text{argmax}_g \mathbb{E}[P_{H \rightarrow \infty}^\gamma], & \text{si } \gamma > 0 \\ g_{\gamma, \infty}^*(b) = \text{argmax}_g \mathbb{E}[\ln(P_{H \rightarrow \infty})], & \text{si } \gamma = 0 \\ g_{\gamma, \infty}^*(b) = \text{argmax}_g \mathbb{E}[-P_{H \rightarrow \infty}^\gamma], & \text{si } \gamma < 0 \end{cases}$$

A continuación, se deduce la forma algebraica de la esperanza para cada uno de los casos, para posteriormente proceder a abordar el problema de optimización:

Utilidad Esperada

Nivel de riesgo $\gamma = 0$:

Para el caso en que el apostador valora logarítmicamente su patrimonio, se dedujo en los desarrollos previos que dicha valoración converge a una distribución normal, y por lo tanto se tiene el siguiente resultado:

$$\lim_{n \rightarrow \infty} E_g[\ln(P_{n+1})] = \lim_{n \rightarrow \infty} (n \cdot \mu_g)$$

Por su relevancia, se destaca una aproximación para el percentil de la esperanza luego de n rondas jugadas, el cual converge en el límite $n \rightarrow \infty$:

$$\text{Prob}(\ln(P_{n+1}) \leq n \cdot \mu_g) \approx 0,5$$

Nivel de riesgo $\gamma > 0$:

En este caso el jugador valora el patrimonio como P_n^γ , y dado que en el límite P_{n+1}^γ se comporta igual que una multiplicación de n variables aleatorias independientes e idénticamente distribuidas con distribución $(1 + g^V \cdot X_{\theta^*}^V)^\gamma$, se concluye que la esperanza de P_{n+1}^γ se comporta de la siguiente manera:

$$\lim_{n \rightarrow \infty} E_g [P_{n+1}^\gamma] = \lim_{n \rightarrow \infty} E_g^n [(1 + g^V \cdot X_{\theta^*}^V)^\gamma]$$

Por su relevancia, se destaca una aproximación para el percentil de la esperanza luego de n rondas jugadas, el cual converge en el límite $n \rightarrow \infty$, y que llamaremos λ_n :

$$\text{Prob} (P_{n+1}^\gamma \geq E_g^n [(1 + g^V \cdot X_{\theta^*}^V)^\gamma]) \approx \lambda_n = \Phi \left(\frac{\sqrt{n} \cdot \gamma \cdot \sigma_g}{2} \right)$$

Donde:

$\gamma > 0$.

$\lim_{n \rightarrow \infty} \lambda_n = 1$.

* Aquí se ha usado el hecho de que $\lim_{n \rightarrow \infty} E_g [P_{n+1}^\gamma] = \exp \left(n \cdot \gamma \cdot \sigma_g + \frac{n \cdot \gamma^2 \cdot \sigma_g^2}{2} \right)$, por tratarse de una distribución LogNormal.

Nivel de riesgo $\gamma \leq 0$:

Este caso es similar al caso en que $\gamma > 0$, diferenciándose solamente en el signo. Por lo

tanto, se concluye lo siguiente:

$$\lim_{n \rightarrow \infty} E_g [-P_{n+1}^\gamma] = \lim_{n \rightarrow \infty} -E_g^n [(1 + g^V \cdot X_{\theta^*}^V)^\gamma]$$

Por su relevancia, se destaca una aproximación para el percentil de la esperanza luego de n rondas jugadas, la cual converge en el límite $n \rightarrow \infty$, y que llamaremos δ_n :

$$\text{Prob}(-P_{n+1}^\gamma \leq -E_g^n [(1 + g^V \cdot X_{\theta^*}^V)^\gamma]) \approx \delta_n = 1 - \lambda_n$$

Donde:

$$\gamma < 0.$$

$$\lim_{n \rightarrow \infty} \delta_n = 0.$$

**Un aspecto interesante del percentil de la utilidad esperada $E[r_\gamma(P_{H \rightarrow \infty})]$ es que se va a 0 para $\gamma < 0$ y a 1 para $\gamma > 0$, mientras que para $\gamma = 0$ este converge a 0.5. Esto refuerza el concepto de γ como un indicador de la inclinación al riesgo, siendo $\gamma = 0$ el nivel de riesgo neutro.*

Optimización de la Política

Ahora que hemos expresado algebraicamente la esperanza de $r_\gamma(P_{H \rightarrow \infty})$, procedemos a formular y resolver el problema de optimización $O_{\gamma, H \rightarrow \infty}$:

Optimización para $\gamma > 0$:

Debemos encontrar la política g^b que solucione el siguiente problema:

$$g_{\gamma, \infty}^*(b) = \text{argmax}_g \left\{ \lim_{n \rightarrow \infty} \mathbb{E}_g^n [(1 + g^V \cdot X_{\theta^*}^V)^\gamma] \right\}$$

Con:

$$b \in \Omega_{c, m}$$

Notamos que maximizar $\mathbb{E}^n [(1 + g^V \cdot X_{\theta^*}^V)^\gamma]$ es equivalente con maximizar $\mathbb{E} [(1 + g^V \cdot X_{\theta^*}^V)^\gamma]$:

$$g_{\gamma, \infty}^*(b) = \text{argmax}_g \left\{ \mathbb{E} [(1 + g^V \cdot X_{\theta^*}^V)^\gamma] \right\}$$

$$g_{\gamma, \infty}^*(b) = \text{argmax}_g \left\{ \sum_{v \in \Omega_{c, m}} p_v \cdot \mathbb{E} [(1 + g^v \cdot X_{\theta^*}^v)^\gamma] \right\}$$

El problema de maximización es independiente para cada $v \in \Omega_{c,m}$, por lo tanto, se deduce que la política que maximiza $r_\gamma(P_{H \rightarrow \infty})$ será aquella que a partir de la baraja de origen b de cada ronda, maximice el valor de $\mathbb{E}[(1 + g^b \cdot X_{\theta^*}^b)^\gamma]$:

$$g_{\gamma,\infty}^*(v) = \operatorname{argmax}_g \{ \mathbb{E}[(1 + g^v \cdot X_{\theta^*}^v)^\gamma] \}, \forall v \in \Omega_{c,m}$$

$$g_{\gamma,\infty}^*(v) = \operatorname{argmax}_g \left\{ \sum_{x \in \text{PagosBJ}} \operatorname{Prob}(X_{\theta^*}^v = x) \cdot (1 + g^v \cdot x)^\gamma \right\}, \quad \forall v \in \Omega_{c,m}$$

Con:
 $\gamma > 0$.

Luego, si derivamos la expresión respecto a la apuesta g^v , encontramos la condición que debe cumplir $g_{\gamma,\infty}^*(b)$ para maximizar $r_\gamma(P_{H \rightarrow \infty})$, la cual aplica para cada baraja de origen que puede darse en una sesión de apuestas $v \in \Omega_{c,m}$:

$$\sum_{x \in \text{PagosBJ}} \operatorname{Prob}(X_{\theta^*}^v = x) \cdot \gamma \cdot (1 + g^v \cdot x)^{\gamma-1} \cdot x = 0, \quad \forall v \in \Omega_{c,m}$$

Debido a que solo existen 6 posibles retornos en *PagosBJ*, y ya se logró calcular $\operatorname{Prob}(X_{\theta^*}^v = x)$ a través del programa desarrollado en la primera parte de esta investigación, solucionar esta ecuación es trivial computacionalmente. Un caso particular es cuando $\gamma = 1$, para el cual la política parcial óptima que resuelve el problema de optimización $O_{1,\infty}$ se puede hallar explícitamente, y tiene la siguiente forma:

$$g_{1,\infty}^*(b) = \begin{cases} 0,5, & \text{si } \mathbb{E}[X_{\theta^*}^b] > 0 \\ 0, & \text{si } \mathbb{E}[X_{\theta^*}^b] \leq 0 \end{cases}$$

Este resultado es paradójico, pues a pesar de ser la política de máximo retorno posible, lleva casi seguramente a la ruina como se explicó previamente (pues se apuesta 0.5), y la probabilidad de obtener efectivamente un retorno igual al retorno esperado se va a 0. Esta paradoja surge por valorar el dinero de acuerdo con su valor monetario $r_1(p) = p$, y es análoga a la famosa *paradoja de san Petersburgo*, formulada originalmente por Nicolaus Bernoulli. A raíz de esta problemática surgen posibles soluciones al problema, siendo una de ellas la utilización de una función de utilidad sobre las ganancias, tal como se ha propuesto en el presente trabajo. Este resultado sugiere que los valores con sentido práctico para γ probablemente sean menores a 1.

Finalmente, una propiedad importante de la política óptima $g_{\gamma, \infty}^*(b)$, consiste en que esta determina *no apostar* cuando $\mathbb{E}[X_{\theta^*}^b] \leq 0$, para cualquier $\gamma > 0$. Esto se deduce al observar que la derivada de $\mathbb{E}[(1 + g^v \cdot X_{\theta^*}^v)^\gamma]$ respecto a la apuesta g^v , evaluada en $g^v = 0$, solo es positiva si $\mathbb{E}[X_{\theta^*}^b]$ es positiva:

$$\left. \frac{d\mathbb{E}[(1 + g^v \cdot X_{\theta^*}^v)^\gamma]}{dg^v} \right|_{g=0} = \sum_{x \in \text{PagosBJ}} \text{Prob}(X_{\theta^*}^v = x) \cdot \gamma \cdot (1 + 0 \cdot x)^{\gamma-1} \cdot x = \gamma \cdot \mathbb{E}[X_{\theta^*}^v]$$

Por lo tanto, la política óptima $g_{\gamma, \infty}^*(b)$ es mayor a 0 solo si $\mathbb{E}[X_{\theta^*}^b] > 0$, pues cuando esto ocurre la derivada de $\mathbb{E}[\ln(1 + g^b \cdot X_{\theta^*}^b)]$ es positiva en $g^v = 0$:

Optimización para $\gamma < 0$:

Veremos que este caso es análogo al caso en que γ es positivo. Debemos encontrar la política g^b que solucione el siguiente problema:

$$g_{\gamma, \infty}^*(b) = \operatorname{argmax}_g \left\{ \lim_{n \rightarrow \infty} -\mathbb{E}_g^n [(1 + g^V \cdot X_{\theta^*}^V)^\gamma] \right\}$$

Con:

$$b \in \Omega_{c, m}$$

Esto es equivalente con minimizar el valor de $\mathbb{E}_g^n [(1 + g^V \cdot X_{\theta^*}^V)^\gamma]$:

$$g_{\gamma, \infty}^*(b) = \operatorname{argmin}_g \left\{ \lim_{n \rightarrow \infty} \mathbb{E}_g^n [(1 + g^V \cdot X_{\theta^*}^V)^\gamma] \right\}$$

Por lo tanto, debido a que la expresión es análoga al caso en que γ es positivo pero esta vez se minimiza en vez de maximizar, se concluye que la condición que debe cumplir $g_{\gamma, \infty}^*(b)$ es la misma que para el caso $\gamma > 0$, la cual aplica para cada baraja de origen que puede darse en una sesión de apuestas $v \in \Omega_{c, m}$:

$$\sum_{x \in \text{PagosBJ}} \text{Prob}(X_{\theta^*}^v = x) \cdot \gamma \cdot (1 + g^v \cdot x)^{\gamma-1} \cdot x = 0, \quad \forall v \in \Omega_{c, m}$$

Por último, siguiendo un procedimiento algebraico similar al caso $\gamma > 0$, la política óptima $g_{\gamma, \infty}^*(b)$ conserva la propiedad de ser mayor a 0 solo si $\mathbb{E}[X_{\theta^*}^b] > 0$.

Optimización para $\gamma = 0$:

Buscamos solucionar $O_{0,\infty}$. Es decir, debemos encontrar la política g^b que solucione el siguiente problema:

$$g_{0,\infty}^*(b) = \operatorname{argmax}_g \left\{ \lim_{n \rightarrow \infty} (n \cdot \mu_g) \right\}$$

Con:

$$b \in \Omega_{c,m}$$

Notando que maximizar $(n \cdot \mu_g)$ es equivalente a maximizar μ_g , anotamos el problema de maximización como sigue:

$$g_{0,\infty}^*(b) = \operatorname{argmax}_g \{\mu_g\}$$

$$g_{0,\infty}^*(b) = \operatorname{argmax}_g \left\{ \sum_{v \in \Omega_{c,m}} p_v \cdot \mathbb{E}[\ln(1 + g^v \cdot X_{\theta^*}^v)] \right\}$$

Similar al caso $\gamma \neq 0$, el problema de maximización es independiente para cada $v \in \Omega_{c,m}$:

$$g_{0,\infty}^*(v) = \operatorname{argmax}_g \{\mathbb{E}[\ln(1 + g^v \cdot X_{\theta^*}^v)]\}, \quad \forall v \in \Omega_{c,m}$$

Luego, si desarrollamos $\mathbb{E}[\ln(1 + g^v \cdot X_{\theta^*}^v)]$ y derivamos la expresión respecto a la apuesta g^v , encontramos la condición que debe cumplir $g_{0,\infty}^*(b)$ para maximizar $r_0(P_{H \rightarrow \infty})$, la cual aplica para cada baraja de origen que puede darse en una sesión de apuestas $v \in \Omega_{c,m}$:

$$\sum_{x \in \text{PagosBJ}} \frac{\operatorname{Prob}(X_{\theta^*}^v = x) \cdot x}{1 + g^v \cdot x} = 0, \quad \forall v \in \Omega_{c,m}$$

Debido a que solo existen 6 posibles retornos en *PagosBJ*, y ya logramos calcular $\operatorname{Prob}(X_{\theta^*}^v = x)$ a través del programa desarrollado en la primera parte de esta investigación, solucionar esta ecuación es trivial computacionalmente. Similar al caso $\gamma \neq 0$, la política óptima $g_{0,\infty}^*(b)$ también determina *no apostar* cuando $\mathbb{E}[X_{\theta^*}^b] \leq 0$. Esto se deduce al observar que la derivada de $\mathbb{E}[\ln(1 + g^v \cdot X_{\theta^*}^v)]$ respecto a la apuesta g^v , evaluada en $g^v = 0$, solo es positiva si $\mathbb{E}[X_{\theta^*}^b]$ es positiva:

$$\left. \frac{d\mathbb{E}[\ln(1 + g^v \cdot X_{\theta^*}^v)]}{dg^v} \right|_{g=0} = \sum_{x \in \text{PagosBJ}} \frac{\text{Prob}(X_{\theta^*}^v = x) \cdot x}{1 + 0 \cdot x} = \mathbb{E}[X_{\theta^*}^v]$$

Por lo tanto, la política óptima $g_{0,\infty}^*(b)$ es mayor a 0 solo si $\mathbb{E}[X_{\theta^*}^b] > 0$, pues cuando esto ocurre la derivada de $\mathbb{E}[\ln(1 + g^b \cdot X_{\theta^*}^b)]$ es positiva en $g^b = 0$. Esta propiedad se mantuvo para todo nivel de riesgo γ , demostrando ser una propiedad general respecto de la política óptima $g_{\gamma,\infty}^*(b)$.

7. Tercera Parte: Exploración de Soluciones

Si bien la optimización de la política con información completa es compleja y requiere redes neuronales, en el presente trabajo se exploraron soluciones para estructuras algebraicas específicas de la función de apuesta. Estas estructuras incorporan las variables de estado y parámetros libres ajustables, de tal manera de poseer flexibilidad para modelar una familia de funciones de apuesta. Luego, dichos modelos algebraicos fueron propuestos como política para 4 problemas de optimización, definidos por sesiones de apuestas compuestas por $H \in \{100, 1000\}$ rondas, restringidos por un parámetro $R \in \{0, 1\}$ que se refiere al *risk of ruin* (*ROR*), y que incorporan una función de utilidad correspondiente a la mediana del patrimonio:

Optimización de la Mediana:

$$\pi_H^*(s) = \operatorname{argmax}_g \{r(P_H)\}$$

sujeto a: $\begin{cases} \pi_H^* \in [0, 0,5], & \text{si } R \neq 1 \\ \pi_H^* \in [0, 0,5), & \text{si } R = 0 \end{cases}$

Con:

$r(P_H) = \text{Mediana}(P_H)$: Función de Utilidad.

$H \in \{100, 1000\}$: Número de Rondas de la sesión.

$R \in \{0, 1\}$: Parámetro que permite o no el *ROR*.

La razón de haber escogido la mediana como función de utilidad radica en los resultados teóricos obtenidos en secciones previas para la política parcial, en donde se mostró que resolver el problema de optimización $O_{0,\infty}$ para el nivel de riesgo neutro $\gamma = 0$ era equivalente a maximizar la mediana del patrimonio cuando $H \rightarrow \infty$. Por lo tanto, estableciendo la mediana como función de utilidad se busca mantener la neutralidad de la inclinación al riesgo y asegurar una buena medida de tendencia central, ofreciendo así un contexto más significativo y representativo para los resultados del experimento. Además, se espera dar una mayor perspectiva a la solución $g_{0,\infty}^*(b)$ encontrada para la política parcial, comparando su desempeño respecto de los distintos modelos propuestos para la política.

7.1. Modelos para la Política

Se postularon 4 modelos sencillos para la política, que gozan de buena interpretabilidad, y se presentan a continuación, ordenados desde el más simple al más complejo:

1. **Apuesta Porcentual Plana:** Este modelo para la política de apuestas es el más sencillo de todos. Consiste en que el jugador apuesta un porcentaje fijo igual para cada

ronda de la sesión. Tiene un parámetro libre y su forma es como sigue:

$$\pi_1(b, p, n; w) = w$$

2. **Apuesta Neta Plana:** Este modelo es algebraicamente más complejo, aunque más sencillo conceptualmente. Consiste en que el jugador apuesta un monto fijo para cada ronda de la sesión, cuyo valor se define como un porcentaje del patrimonio inicial. Esta función tiene un parámetro libre, pero también considera el patrimonio de manera inversamente proporcional a la apuesta porcentual. Su forma algebraica es como sigue:

$$\pi_2(b, p, n; e) = \frac{e}{p}$$

3. **Política Parcial dependiente de la Baraja:** Este modelo para la función de apuestas solo considera la baraja b para su formulación, y determina un porcentaje específico asociado a cada baraja que toca en el juego. Se corresponde con la política parcial óptima planteada en secciones previas, y su forma es como sigue:

$$\pi_3(b, p, n) = g_{0,\infty}^*(b)$$

4. **Apuesta Compuesta:** Este modelo fue diseñado específicamente para enriquecer los resultados. Tiene una forma algebraica más compleja que los modelos previos incorporando 3 parámetros libres. Sin embargo, bajo determinadas configuraciones de los parámetros este modelo tiene la flexibilidad de reducirse a cualquiera de los 3 modelos previos, asegurando resultados por lo menos igual de buenos que los demás modelos. Su forma algebraica es como sigue:

$$\pi_4(b, p, n; \alpha, k, c) = k \cdot \pi_3^c \cdot p^\alpha$$

7.2. Metodología

En esta sección se explica el proceso diseñado y los mecanismos utilizados para solucionar el problema de optimización propuesto para los 4 modelos que definen la política, a partir de los conceptos expuestos en el desarrollo de la presente investigación. Se detallan todos los aspectos relevantes considerados para simular el juego, describiendo la metodología utilizada para simular secuencias de rondas y explicando el proceso de optimización y testeo para los modelos propuestos. Se reitera que durante todo el proceso diseñado se asumen las reglas y condiciones establecidas al comienzo de este trabajo.

7.2.1. Secuencias de Barajas de Origen

Como se mostró en el desarrollo de este trabajo, una sesión de apuestas puede ser modelada como una secuencia de barajas de origen $\{B_n\}_{(0 \leq n)}$, con la respectiva secuencia que define

el desarrollo del patrimonio del jugador $\{P_n\}_{(0 \leq n)}$. Sin embargo, mientras que la evolución del patrimonio sí depende de la política de apuestas, la secuencia de barajas de origen es independiente de esta, y depende solamente de la política de juego del jugador.

Por lo tanto, dado que la función de transición de la secuencia $\{B_n\}_{(0 \leq n)}$ puede ser calculada con el programa desarrollado, se procedió a simular barajas secuencialmente, identificándolas según el juego al que correspondan y su posición dentro de dicho juego. De esta manera, se simularon 5.000 juegos completos, resultando en un total de 257.520 barajas de origen simuladas.

7.2.2. Secuencias de Rondas

Para simular un jugador jugando ronda tras ronda, se utilizaron las barajas de origen previamente simuladas, que componen los 5.000 juegos. Dado que cada una de estas barajas lleva asociada una FMP para el retorno sobre la apuesta del jugador (asumiendo política de juego óptima), la cual es conocida a partir del programa mencionado, luego es posible generar un retorno de manera aleatoria a partir de cada una de las barajas simuladas. Por lo tanto, cada una de las 257.520 barajas de origen simuladas sirve como argumento para generar aleatoriamente un retorno perteneciente a $PagosBJ$, según la FMP asociada.

Si bien en la simulación de los 5.000 juegos la secuencia de barajas generada lleva asociada implícitamente una secuencia de retornos obtenidos por el jugador, se considera esto como un detalle de no mayor relevancia, dado que el retorno obtenido a partir de una cierta baraja de origen solo lleva asociado una correlación con la cantidad de cartas que se extraen para llegar a la próxima baraja de origen. Lo verdaderamente importante de una baraja es la FMP para la próxima baraja que se jugará, independiente del retorno asociado a llegar a dicha baraja.

7.2.3. Optimización de los Modelos y Testeo

Para el proceso de optimización y testeo de los modelos propuesto, se dividió la muestra total de 5.000 juegos simulados en dos conjuntos distintos: datos de entrenamiento y datos de testeo. De estos, 4.000 juegos, que constituyen el 80% de la muestra total, se reservaron para el entrenamiento, mientras que los 1.000 juegos restantes se utilizaron para el testeo. Sin embargo, *la muestra total fue previamente filtrada, de tal manera de contener únicamente aquellas barajas de origen con un retorno esperado positivo para el jugador*. Esta decisión se fundamentó en el desarrollo teórico previo, donde se mostró que la política parcial óptima $\pi_3(b, p, n) = g_{0, \infty}^*(b)$ solo apuesta cuando $E[X_{\theta^*}^b] > 0$. Por lo tanto, se excluyeron de la muestra total todas aquellas barajas de origen que no cumplieran con este criterio, asegurando que todos los modelos propuestos sean optimizados y testeados solo en barajas de origen con retorno esperado positivo.

Optimización:

A partir de los datos de entrenamiento, que corresponden a 4.000 juegos completos filtrados, se procedió a simular sesiones de apuestas para cada uno de los escenarios de estudio de manera independiente. Un escenario de estudio se refiere a la combinación de uno de los modelos propuestos (excluyendo el modelo de política parcial, el cual se obtiene directamente a partir de la muestra total) con una configuración del problema de optimización. Para cada escenario, se generaron un total de 100.000 sesiones de apuestas, cada una compuesta de H barajas de origen, según el problema de optimización. Cada una de estas sesiones se generó seleccionando juegos secuencialmente de manera equiprobable entre los datos de entrenamiento.

El siguiente paso consistió en simular rondas jugadas a partir de las barajas de origen contenidas en cada sesión. Este proceso se llevó a cabo para las 100.000 sesiones simuladas, donde cada baraja de origen contenida en dichas sesiones se utilizó para generar un retorno aleatorio para el jugador, según la FMP de dicha baraja. De esta manera, las secuencias de barajas que definen las sesiones generaron secuencias de retornos para el jugador, pertenecientes a $PagosBJ = [-2, -1, 0, 1, 1.5, 2]$.

Finalmente, los modelos propuestos para la política fueron optimizados para cada uno de los 4 problemas de optimización. A partir de las 100.000 sesiones compuestas por secuencias de retornos (que son retornos sobre la apuesta que se realiza), cada modelo fue inicializado con valores arbitrarios para sus parámetros, los cuales fueron optimizados a través del método SLSQP. Cada configuración de parámetros implica una estrategia de apuestas, que al aplicarse sobre cada secuencia de retornos resulta en un retorno final para cada sesión. Tal como se definió el problema de optimización, el valor objetivo utilizado como criterio de optimalidad fue la mediana del retorno obtenido en las 100.000 sesiones, aplicando la restricción sobre el ROR cuando fuese necesario.

Se reitera por claridad que todo el proceso mencionado fue llevado a cabo para cada escenario de estudio de manera independiente, utilizando siempre la data de entrenamiento compuesta por 4.000 juegos filtrados. Por último, el modelo número 3, “política parcial dependiente de la baraja”, es un caso aparte de los otros modelos, y no requirió de simulaciones para ser optimizado. En cambio, la forma exacta de esta política, $\pi_3(b, p, n) = g_{(0, \infty)}^*(b)$, se obtuvo deduciendo una apuesta fraccionaria para cada baraja v contenida en la muestra filtrada, solucionando la siguiente ecuación, que fue detallada en el desarrollo teórico:

$$\sum_{x \in PagosBJ} \frac{\text{Prob}(X_{\theta^*}^v = x) \cdot x}{1 + g(v) \cdot x} = 0$$

Testeo:

Luego de optimizar cada modelo para cada problema de optimización, se obtuvieron los

modelos optimizados. El testeo de estos modelos fue realizado a partir de los datos de testeo, correspondientes a los 1.000 juegos completos filtrados que no fueron considerados para el entrenamiento. Similar al proceso de optimización, se simularon 100.000 sesiones de apuestas para cada modelo optimizado de manera independiente. Cada una de estas sesiones fue generada de manera análoga a las sesiones de entrenamiento, resultando en sesiones compuestas de H barajas de origen (según el problema de optimización), donde a su vez cada baraja fue utilizada para generar un retorno para el jugador, perteneciente a $PagosBJ = [-2, -1, 0, 1, 1,5, 2]$. Luego, para cada caso, los modelos optimizados fueron aplicados a cada una de las 100.000 sesiones simuladas, alcanzando un retorno final en cada una de ellas. Finalmente, a partir de los 100.000 retornos finales resultantes, se deduce la distribución del retorno asociada a cada modelo optimizado.

8. Resultados

Esta sección detalla los resultados clave del experimento, diferenciando entre aquellos relacionados con la política de juego óptima y los asociados a la optimización de los modelos propuestos para la política de apuestas. Estos resultados se derivan de una muestra de 257.520 barajas de origen simuladas, siguiendo las reglas y condiciones especificadas en la metodología. Este conjunto de datos representa 5.000 juegos completos, desde el inicio con la baraja completa hasta que se alcanza el punto de corte.

Para generar estas barajas, se utilizó la función de transición para la baraja de origen bajo la política de juego óptima, obtenida con el programa desarrollado para optimizar la política. A continuación, se determinó la Función de Masa de Probabilidad (FMP) de los retornos sobre la apuesta del jugador para cada baraja de la muestra, utilizando el mismo software. Esto permitió examinar las características generales del retorno en el juego.

En una fase posterior, se optimizaron los parámetros de los modelos propuestos para la política de apuestas de tal manera de maximizar la mediana de los retornos, a partir de simulaciones de sesiones de apuestas compuestas de $H \in \{100, 1000\}$ rondas, y considerando la posibilidad de llegar a la ruina ($R=1$), o no ($R=0$). Estas sesiones de optimización se basaron en los datos de entrenamiento, correspondientes a 4.000 juegos completos filtrados, los cuales solo contienen barajas de origen con expectativa positiva para el jugador. Finalmente, los modelos optimizados fueron testeados en nuevas sesiones simuladas, esta vez considerando la data de testeo, correspondiente a los 1.000 juegos completos filtrados restantes.

Para facilitar la comprensión, los resultados se han dividido en 3 subsecciones: “Resultados sobre la Política de Juego”, “Resultados sobre la Política de Apuestas” y “Distribuciones del Retorno”.

8.1. Resultados sobre la Política de Juego

8.1.1. Eficiencia de Cómputo

Un primer resultado se refiere a eficiencia computacional del programa propuesto para calcular la política de juego óptima. Como ya se mencionó, el código propuesto en el trabajo “Las Matemáticas del Blackjack” permite computar la política óptima para instancias iniciales o posteriores, y en principio es fácilmente modificable para computar la FMP del retorno de la ronda a partir de la baraja de origen. Sin embargo, el autor declara que su código presenta un costo computacional “enorme”, y se estima que calcular el FMP del retorno de la ronda podría tomar varias horas. Durante el desarrollo de esta investigación, se reformuló el programa citado evitando una gran cantidad de cálculos repetidos y aprovechando relaciones recursivas, logrando una notable mejora en eficiencia computacional. A continuación, se presenta una comparación de los tiempos de ejecución de ambos programas, específicamente de la función *JugadorDecide* (trabajo citado) y *JugadorDecideRapido* (función desarrollada),

que representan el cuello de botella del problema de eficiencia computacional.

JugadorDecideRapido:

El algoritmo desarrollado en esta investigación está específicamente optimizado para la instancia de origen de la ronda, con la finalidad de obtener la FMP del retorno de la ronda. Por esta razón, el tiempo de cómputo de *JugadorDecideRapido* es prácticamente igual independiente de la instancia de ronda que se evalúe, en contraste con *JugadorDecide*. Asumiendo una baraja de origen completa con 8 mazos, el tiempo de ejecución de la función para la instancia de origen de la ronda (y por lo tanto para cualquier instancia intermedia de la ronda) resultó en tan solo 0.7 segundos:

	Tiempo de Cómputo (s)
JugadorDecideRapido(0, 0, 0, 0, 0, b)	0,7

Tabla 1: Tiempo de Cómputo *JugadorDecideRapido*

Con:

$$b = [32, 32, 32, 32, 32, 32, 32, 32, 128]$$

JugadorDecide:

Por su parte, el algoritmo propuesto en el trabajo citado sí realiza una cantidad de cálculos que depende fuertemente de la profundidad de la instancia de ronda evaluada. Debido a su estructura recursiva y no optimizada para cálculos repetidos, *JugadorDecide* toma tiempos significativamente mayores en computar instancias que presentan muchas “ramificaciones”, y computarla para la instancia de origen de la ronda no fue posible. Sin embargo, sí fue posible computarla para instancias intermedias de la ronda, permitiendo hacerse una idea del orden del costo de la función. A continuación, se presenta el tiempo de cómputo de *JugadorDecide* para distintas instancias de ronda, escogidas adecuadamente para que sean representativas del costo computacional de la función según la profundidad de la instancia:

	Tiempo de Cómputo (s)
JugadorDecide(4, 0, 2, 9, 0, b)	285,2
JugadorDecide(10, 0, 2, 11, 1, b)	26,4
JugadorDecide(15, 1, 2, 10, 0, b)	44,5
JugadorDecide(18, 1, 2, 2, 0, b)	302,2

Tabla 2: Tiempo de Cómputo *JugadorDecide*

Con:

$b = [32, 32, 32, 32, 32, 32, 32, 32, 32, 128]$

De estos resultados se extrae que el programa desarrollado en esta investigación representa una mejora de varios ordenes de magnitud en cuanto a costo computacional. Esto permite generar una gran cantidad de barajas de origen con su respectiva FMP, haciendo posible estudiar la rentabilidad del juego en general, y facilitando la solución del problema general de la política de apuestas a través del uso de redes neuronales por refuerzo.

8.1.2. Rentabilidad del Juego en General

1. FMP del Retorno Global del Juego:

Considerando cada baraja de las 257.520 barajas simuladas como equiprobable, y la Función de Masa de Probabilidad (FMP) del retorno asociada a cada una, se calculó FMP del retorno para el jugador en general. Llamando a este retorno como el “retorno global” del juego, y anotándolo simplemente como $X_{\theta^*}^G$, los resultados obtenidos fueron los siguientes:

$$\text{Prob}(X_{\theta^*}^G = x) = \begin{cases} 4,04 \% & , \text{ si } x = -2 \\ 43,93 \% & , \text{ si } x = -1 \\ 8,58 \% & , \text{ si } x = 0 \\ 33,23 \% & , \text{ si } x = 1 \\ 4,52 \% & , \text{ si } x = 1,5 \\ 5,67 \% & , \text{ si } x = 2 \end{cases}$$

En particular se extrae que, si se participa de todas las rondas del juego bajo una política de juego óptima, se espera a la larga *perder el doble de la apuesta inicial* un 4,04% de las veces, *perder un monto igual a la apuesta inicial* un 43,93%, *empatar* un 8,58%, *ganar un monto igual a la apuesta inicial* un 33,23%, *ganar por Blackjack* un 4,52% y *ganar el doble de la apuesta inicial* un 5,67% de las rondas. Se observa que la esperanza del retorno global es $-0,0063 = -0,63\%$, mientras que la desviación estándar resulta en $1,123 = 112,3\%$. La esperanza negativa da luz de la ventaja de la casa, aun cuando se juega bajo una política de juego óptima.

2. Histograma del Retorno Esperado:

A partir de las 257.520 barajas simuladas se generó un histograma del retorno esperado que muestra la frecuencia de los distintos retornos esperados para el jugador. Este histograma exhibe una forma altamente simétrica, similar a una campana compri-

mida, con una media negativa de $-0,63\%$, siendo el juego en general ventajoso para la casa, como ya se mostró.

Sin embargo, es importante destacar que 55.281 de las barajas generadas (21,4% de la muestra total) arrojaron un retorno esperado positivo para el jugador, con un retorno promedio de $0,0082=0,82\%$. Esto indica que, siguiendo la política de juego óptima y considerando la composición de la baraja antes del inicio de cada ronda, apostar puede ser ventajoso en más de una quinta parte de las rondas del juego, con un retorno esperado promedio de $0,82\%$ para el jugador.

Por otro lado, se observa que en general el retorno esperado se muestra concentrado cerca de la media, con una desviación estándar de $0,01$. Como valores de referencia se destaca que para el $\sim 99,5\%$ de las barajas el retorno esperado está comprendido entre -4% y $+4\%$, para el $\sim 91\%$ está comprendido entre -2% y $+2\%$, y para el $\sim 61\%$ de los casos es mayor a -1% y menor a 1% . A continuación, visualizamos el histograma del retorno esperado asociado a las 257.520 barajas simuladas:

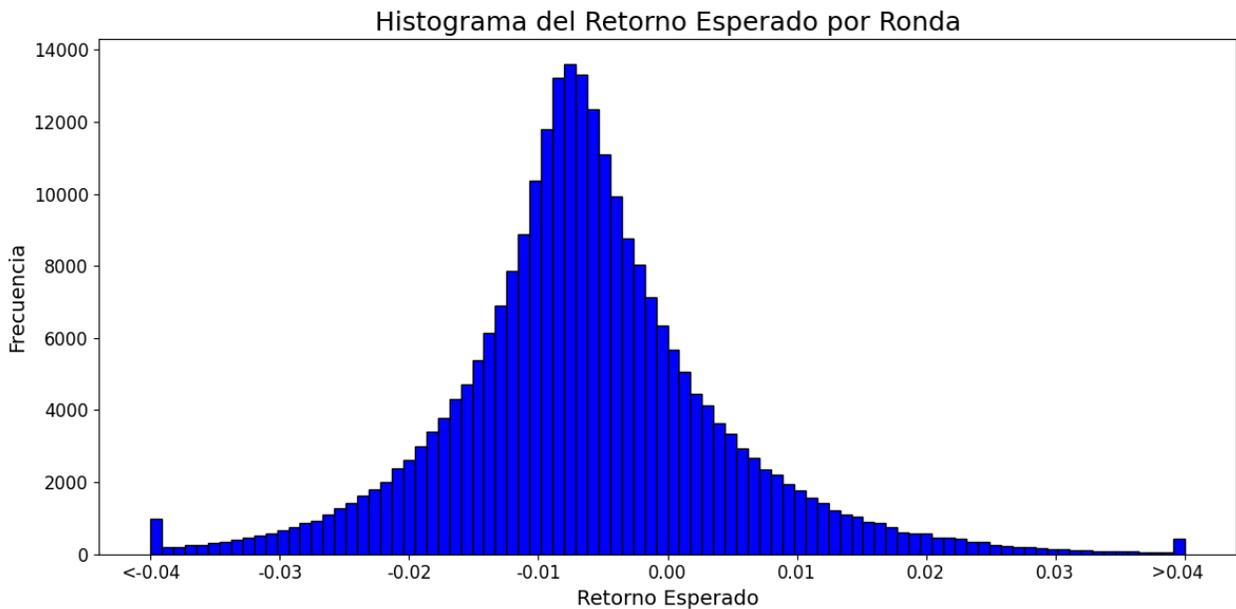


Figura 1: Histograma del Retorno Esperado por Ronda.

3. Histograma Desviación Estándar del Retorno:

El análisis de las barajas simuladas incluyó el cálculo de la desviación estándar del retorno para cada una, resultando en un histograma que refleja las frecuencias observadas. Un detalle notable en la distribución resultante es su asimetría pronunciada, caracterizada por dos picos principales de concentración separados por un valle menos poblado. Específicamente, el punto de mínima concentración del valle ocurre para una

desviación de aproximadamente $\approx 1,13$, la cual coincide con el percentil $\sim 79\%$ de la distribución. Considerando que el $21,4\%$ de las rondas presentan retorno esperado positivo bajo la política de juego óptima, se sugiere que dicha ventaja está correlacionada con resultados de mayor volatilidad como obtener blackjack, doblar o abrir, lo que explicaría la asimetría de la gráfica. A continuación, visualizamos el histograma de las desviaciones estándar del retorno, para los 257.520 mazos simulados:

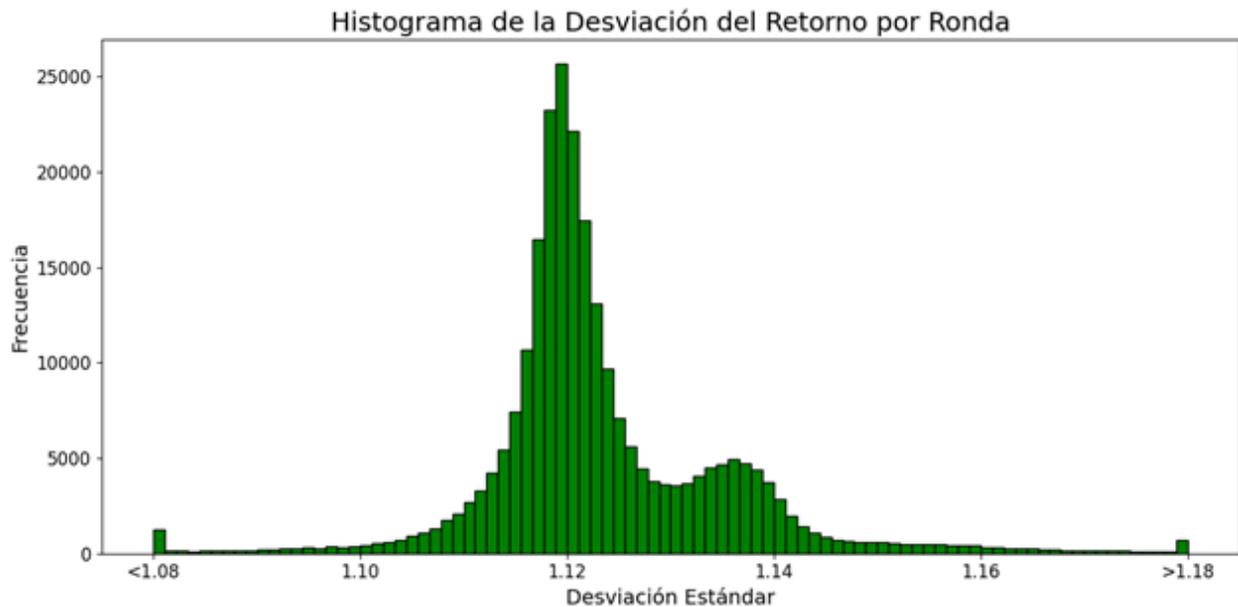


Figura 2: Histograma de la Desviación Estándar del Retorno por ronda.

4. Correlación Retorno Esperado vs Desviación Estándar:

Para analizar la relación entre el retorno esperado de una ronda y su desviación estándar, se elaboró un gráfico de dispersión que considera ambos parámetros. El análisis reveló un coeficiente de correlación significativo de $0,755$, lo cual apoya la hipótesis de que la política de juego óptima logra una ventaja sobre la casa a través de acciones o eventos de mayor volatilidad, como obtener un blackjack, abrir la mano o doblar la apuesta. A continuación, se presenta el gráfico de dispersión que muestra la correlación entre el retorno esperado y la desviación estándar, donde se ha optado por excluir los valores extremos anómalos con ánimos de facilitar la interpretación visual de los datos:

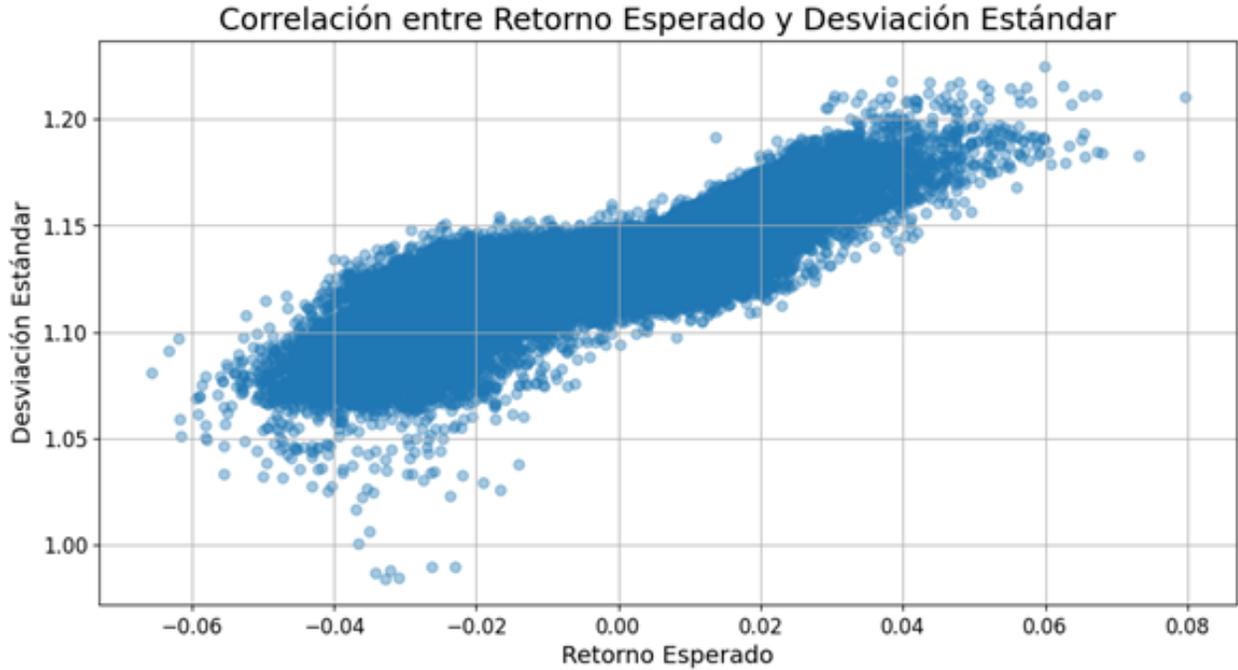


Figura 3: Correlación entre el Retorno Esperado y la Desviación Estándar

8.2. Resultados sobre la Política de Apuestas

En esta sección se muestran los resultados obtenidos respecto al problema de optimización para la política de apuestas, que consiste en maximizar la mediana de los retornos. Cada modelo fue optimizado independientemente a partir de la data de entrenamiento, ajustando sus parámetros libres según 100.000 sesiones de apuestas simuladas, considerando los parámetros H (número de rondas de la sesión de apuestas) y R (parámetro que se refiere a la posibilidad de llegar a la ruina. Si $R = 0$, no se permite llegar a la ruina. Si $R = 1$, no hay restricción) que definen el problema de optimización.

A continuación, se presentan los modelos optimizados resultantes, estructurando según el tipo de modelo algebraico para la política y los parámetros asociados al problema de optimización.

8.2.1. Apuesta Porcentual Plana

Este modelo para la política de apuestas es el más sencillo de todos. Consiste en que el jugador apuesta un porcentaje fijo de su patrimonio igual para cada ronda de la sesión (siempre que la ronda tenga un retorno esperado positivo). El valor de w^* es independiente del valor del parámetro R del problema de optimización (es imposible llegar a la ruina con una apuesta fraccionaria menor a 1), y solo depende del valor de H :

- Problema de Optimización: ($H = 100$, R) : Luego del proceso de optimización, se ob-

tuvo que la apuesta porcentual plana óptima para sesiones compuestas por 100 rondas corresponde a $w^* = 0,69\%$:

$$\pi_1^*(b, p, n; w^*) = w^* = 0,0069$$

- Problema de Optimización: ($H = 1000, R$) :

Para el caso en que las sesiones se componen de 1000 rondas, se obtuvo que la apuesta porcentual plana óptima corresponde a $w^* = 0,66\%$. Este valor difiere muy poco el obtenido para el caso en que $H = 100$, lo cual era esperado a partir del desarrollo teórico:

$$\pi_1^*(b, p, n; w^*) = w^* = 0,0066$$

8.2.2. Política Parcial dependiente de la Baraja

Este modelo para la política de apuestas, $\pi_3(b, p, n) = g_{0,\infty}^*(b)$, se corresponde con la política parcial óptima estudiada en secciones previas, y solo considera la baraja de origen b como variable para determinar la apuesta. Luego de optimizar la apuesta fraccionaria asociada a cada una de las 55.281 barajas de la muestra filtrada (barajas con retorno esperado positivo), lo cual es un proceso independiente de los parámetros H y R del problema de optimización, se obtuvo el siguiente histograma con la frecuencia de las apuestas óptimas:

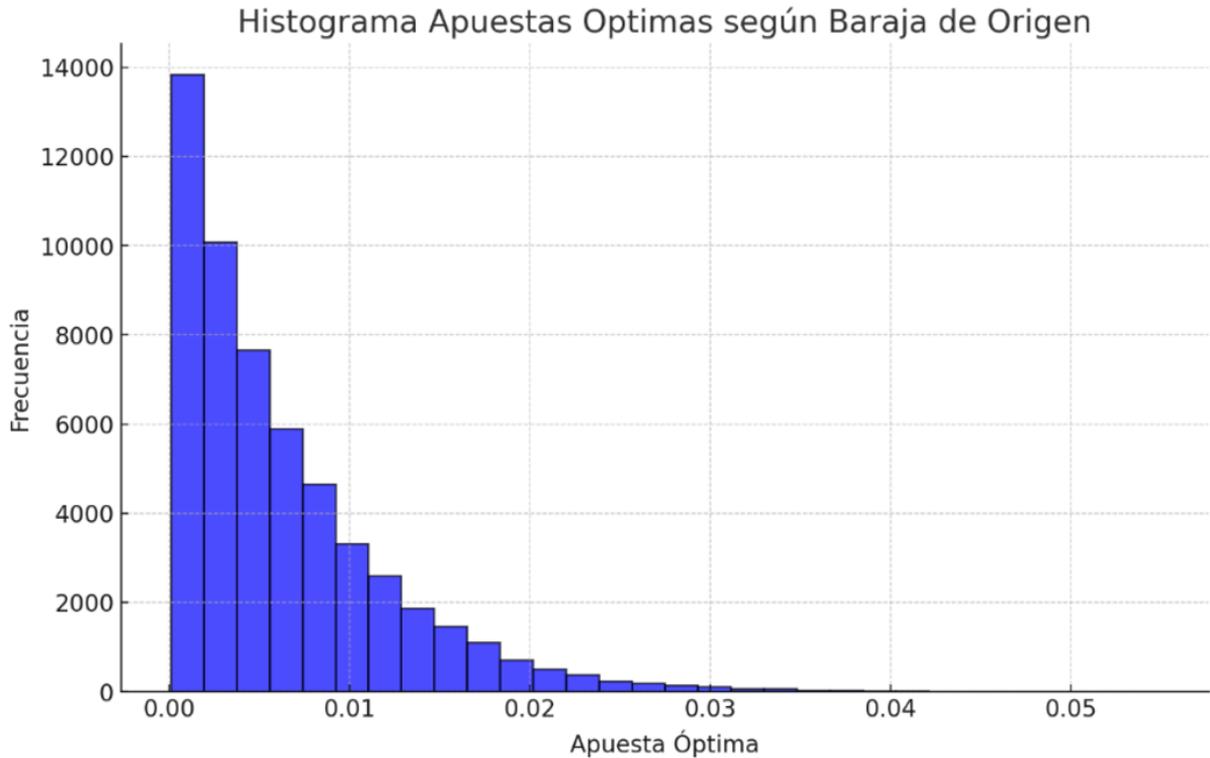


Figura 4: Histograma de la frecuencia de las Apuestas Óptimas.

A partir de este análisis se halló que para un 50% de las barajas con retorno positivo la apuesta óptima es menor al 0,45% del patrimonio. Por otro lado, se observó que para un 79% de las barajas la apuesta óptima es menor al 1%, y que solo para un 4% de las mismas la apuesta óptima supera el 2% del patrimonio. Respecto a la relación entre la apuesta dictada por la política parcial óptima para una baraja v , y el retorno esperado de la misma, $\mathbb{E}[X_{\theta^*}^v]$, se ha elaborado el siguiente gráfico de dispersión:

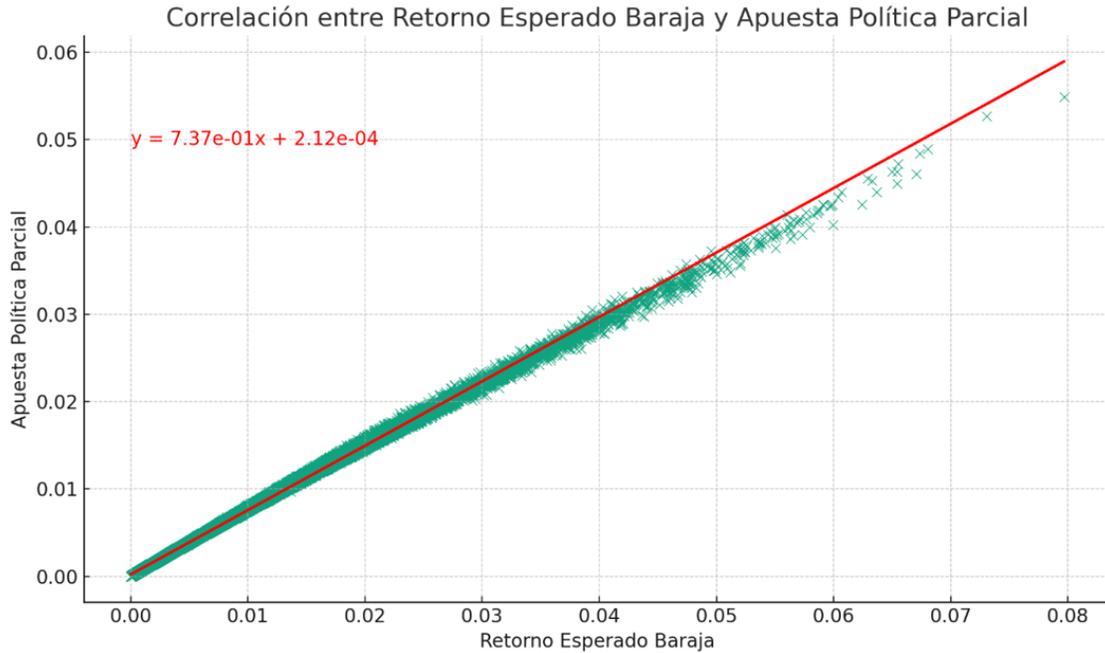


Figura 5: Correlación entre Retorno Esperado y la Apuesta Política Parcial.

De aquí se extrae que el coeficiente de correlación entre ambos valores alcanzó un 0,9991. Esto sugiere una relación lineal entre ambos valores, que tras realizar el ajuste correspondiente, resultó como sigue:

$$g_{0,\infty}^*(v) = 0,737 \cdot \mathbb{E}[X_{\theta^*}^v] + 0,0002$$

Esta aproximación tiene un error absoluto menor a 0,0003 para el $\sim 80\%$ de las barajas filtradas, y menor a 0,001 para el $\sim 98\%$ de las mismas, demostrando ser una excelente aproximación para calcular el valor de la política parcial óptima en términos del retorno esperado de la baraja de origen.

Finalmente, a partir de la FMP de cada una de las barajas filtradas y la apuesta óptima asociada a cada una, se deducen los valores de μ_g (esperanza del logaritmo del patrimonio. Es el valor objetivo que fue maximizado en el proceso de optimización de la política parcial,

lo cual es equivalente con maximizar la mediana de los retornos) y σ_g . El cálculo de estos valores permite estimar la FMP del retorno para el jugador en términos de la cantidad de rondas jugadas (siguiendo la política parcial óptima), y su valor resultó como sigue:

$$\mu_g=0,00004908$$

$$\sigma_g=0,009912257$$

8.2.3. Apuesta Plana Neta

- Problema de Optimización: ($H = 100, R = 0$) :

Luego del proceso de optimización, se obtuvo que la apuesta plana neta óptima (para barajas de origen filtradas, con retorno esperado positivo) corresponde a $e^* = 1,92\%$ del patrimonio inicial. Esto resulta en apostar un monto neto igual a 0,0192 en cada ronda cuyo retorno esperado es positivo, a partir de un patrimonio inicial 1:

$$\pi_2^*(b, p, n; e^*) = \frac{e^*}{p} = \min \left(\frac{0,0192}{p}, 0,5 \right)$$

- Problema de Optimización: ($H = 100, R = 1$) :

Para el caso en que no hay restricción respecto a la ruina, se obtuvo que la apuesta plana neta óptima corresponde a $e^* = 7,95\%$ del patrimonio inicial. Esto resulta en apostar un monto neto igual a 0,0795 en cada ronda cuyo retorno esperado es positivo, a partir de un patrimonio inicial 1. Este valor es más de 4 veces mayor al obtenido para el caso ($H = 100, R = 0$), resaltando un mayor nivel de agresividad del modelo cuando se le permite llegar a la ruina:

$$\pi_2^*(b, p, n; e^*) = \frac{e^*}{p} = \min \left(\frac{0,0795}{p}, 0,5 \right)$$

- Problema de Optimización: ($H = 1000, R = 0$) :

Considerando que la sesión de apuestas se compone de 1000 rondas y que no se permite llegar a la ruina, se obtuvo que la apuesta plana neta óptima corresponde a $e^* = 0,79\%$ del patrimonio inicial. Esto resulta en apostar un monto neto igual a 0,0079 en cada ronda cuyo retorno esperado es positivo, a partir de un patrimonio inicial 1. Este valor es menor que la mitad del valor obtenido para el caso ($H = 100, R = 0$), lo cual es esperable, debido a que la sesión de apuestas se compone de una mayor cantidad de rondas, y se mantiene la restricción sobre el riesgo de ruina:

$$\pi_2^*(b, p, n; e^*) = \frac{e^*}{p} = \min \left(\frac{0,0079}{p}, 0,5 \right)$$

- Problema de Optimización: ($H = 1000, R = 1$) :

Finalmente, para el caso en que la sesión de apuestas se compone de 1000 rondas y

no existe restricción respecto a llegar a la ruina, se obtuvo que la apuesta plana neta óptima corresponde a $e^* = 2,62\%$ del patrimonio inicial. Esto resulta en apostar un monto neto igual a 0,0262 en cada ronda cuyo retorno esperado es positivo, a partir de un patrimonio inicial 1. Este valor es aproximadamente un tercio del valor obtenido para el caso $(H = 100, R = 1)$, resaltando un mayor nivel de agresividad del modelo cuando la sesión se compone de menos rondas y no se restringe sobre el riesgo de ruina:

$$\pi_2^*(b, p, n; e^*) = \frac{e^*}{p} = \min\left(\frac{0,0262}{p}, 0,5\right)$$

8.2.4. Apuesta Compuesta

- Problema de Optimización: $(H = 100, R = 0)$:

Luego del proceso de optimización, se obtuvieron los valores de los 3 parámetros que definen el modelo de apuesta compuesta. El modelo optimizado establece apuestas decrecientes respecto al valor del patrimonio actual p , y crecientes respecto a la apuesta asociada a la política parcial óptima.

$$\pi_4^*(b, p, n; \alpha^*, k^*, c^*) = k^* \cdot \pi_3^{c^*} \cdot p^{\alpha^*} = \min(0,0407 \cdot \pi_3^{0,1626} \cdot p^{-1,3126}, 0,5)$$

A modo de mejorar la visualización de esta estrategia, se muestra la gráfica de las apuestas fraccionarias que realiza, representadas con colores, en función de valores para el patrimonio y para la política parcial π_3 (se consideraron rangos para π_3 y p de tal manera de favorecer la visualización del comportamiento del modelo optimizado):

Gráfico de $\pi_4^* = \min(0.0407 \cdot \pi_3^{0.1626} \cdot p^{-1.3126}, 0.5)$

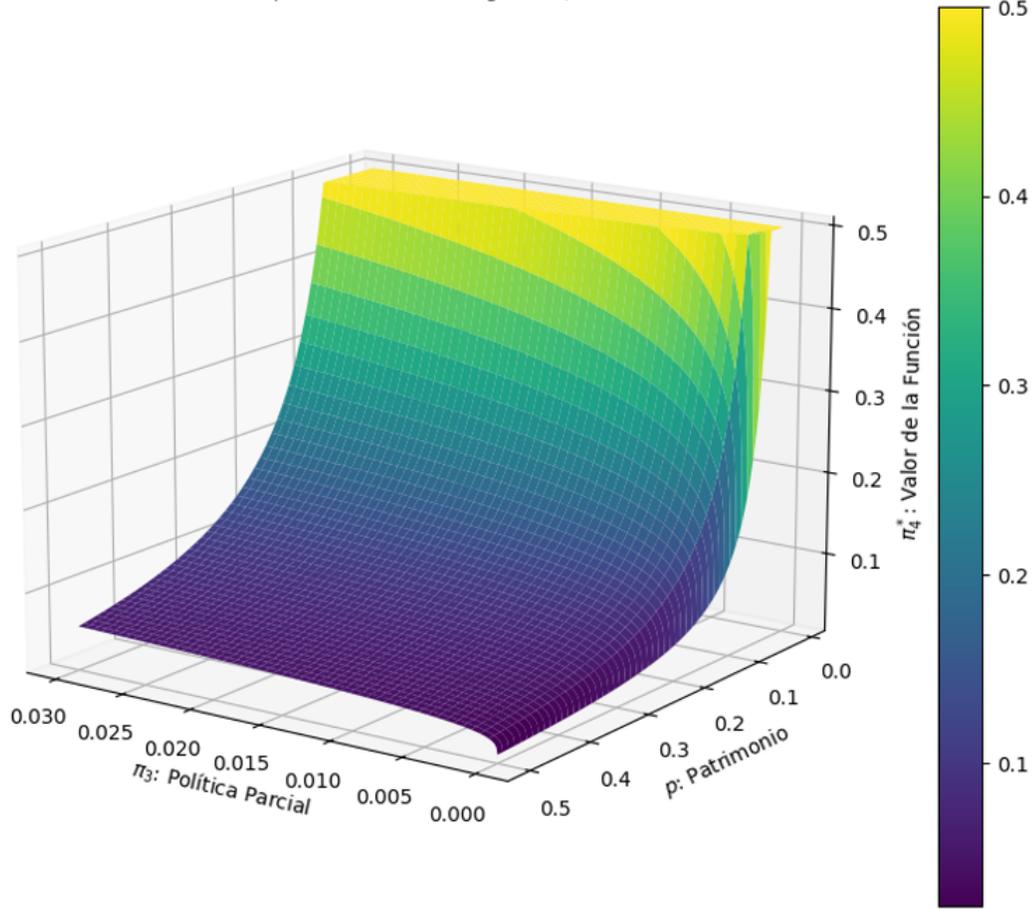


Figura 6: Modelo Compuesto Optimizado para ($H=100, R=0$)

- Problema de Optimización: ($H = 100, R = 1$) :

En comparación al problema de optimización restringiendo respecto al riesgo de ruina, el modelo compuesto optimizado sin restricción resultó marcadamente más agresivo, apostando lo máximo permitido (50 % del patrimonio) siempre que el patrimonio sea menor a $\sim 1,3$. Por último, se obtuvo un valor de $c^* = 0,0001$, lo cual implica que $\pi_3^{0,0001} \approx 1$. Por lo tanto, el modelo optimizado ignora la baraja de origen para determinar la apuesta fraccionaria, y considera solo el valor de su patrimonio actual p :

$$\pi_4^*(b, p, n; \alpha^*, k^*, c^*) = k^* \cdot \pi_3^{c^*} \cdot p^{\alpha^*} = \min(14,43 \cdot \pi_3^{0,0001} \cdot p^{-12,4667}, 0,5)$$

A modo de mejorar la visualización de esta estrategia, se muestra la gráfica de las apuestas fraccionarias que realiza, en función del patrimonio p (se consideró la simplificación $\pi_3^{0,0001} = 1$, con el fin de dejar la estrategia dependiente de una única variable y visualizar mejor su comportamiento):

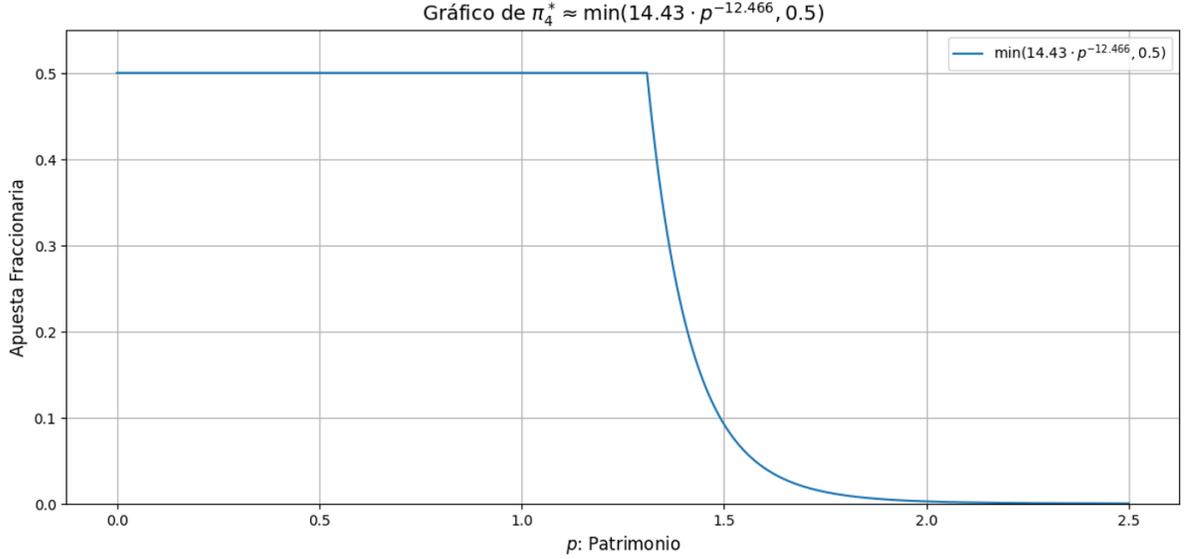


Figura 7: Modelo Compuesto Optimizado para ($H=100, R=1$)

- Problema de Optimización: ($H = 1000, R = 0$) :

Para sesiones de 1000 rondas restringiendo respecto al riesgo de ruina, el modelo optimizado se comporta similar al caso ($H = 100, R = 0$), apostando crecientemente respecto a la política parcial y decrecientemente respecto al valor del patrimonio:

$$\pi_4^*(b, p, n; \alpha^*, k^*, c^*) = k^* \cdot \pi_3^{c^*} \cdot p^{\alpha^*} = \min(0,0243 \cdot \pi_3^{0,1772} \cdot p^{-0,5851}, 0,5)$$

Sin embargo, el modelo optimizado para el caso ($H = 100, R = 0$) resultó más agresivo que para el caso ($H = 1000, R = 0$). Al calcular el cociente entre ambos modelos optimizados (caso $H = 100$ en el numerador), se obtiene $\sim 1,7 \cdot \pi_3^{-0,01} \cdot p^{-0,72}$. Notando el valor de $\pi_3^{-0,01}$ es mayor a 1 y a lo más es $\sim 1,1$ (para $\pi_3 \approx 0,00006$), luego se tiene que el cociente es mayor a 1 siempre que el patrimonio del jugador sea menor a $\sim 2,4$. De aquí se extrae que el modelo compuesto para el caso ($H = 100, R = 0$) apuesta más que para el caso ($H = 1000, R = 0$) casi con seguridad, pues para ambos valores de H alcanzar un patrimonio de 2,4 durante la sesión resulta un evento más bien escaso (ver “Distribuciones del Retorno”). Este resultado es consistente con lo esperado y sugiere que la metodología para optimizar las políticas fue exitosa.

A modo de mejorar la visualización de esta estrategia, se muestra la gráfica de las apuestas fraccionarias que realiza, representadas con colores, en función de valores para el patrimonio y para la política parcial π_3 (se consideraron rangos para π_3 y p de tal manera de favorecer la visualización del comportamiento del modelo optimizado):

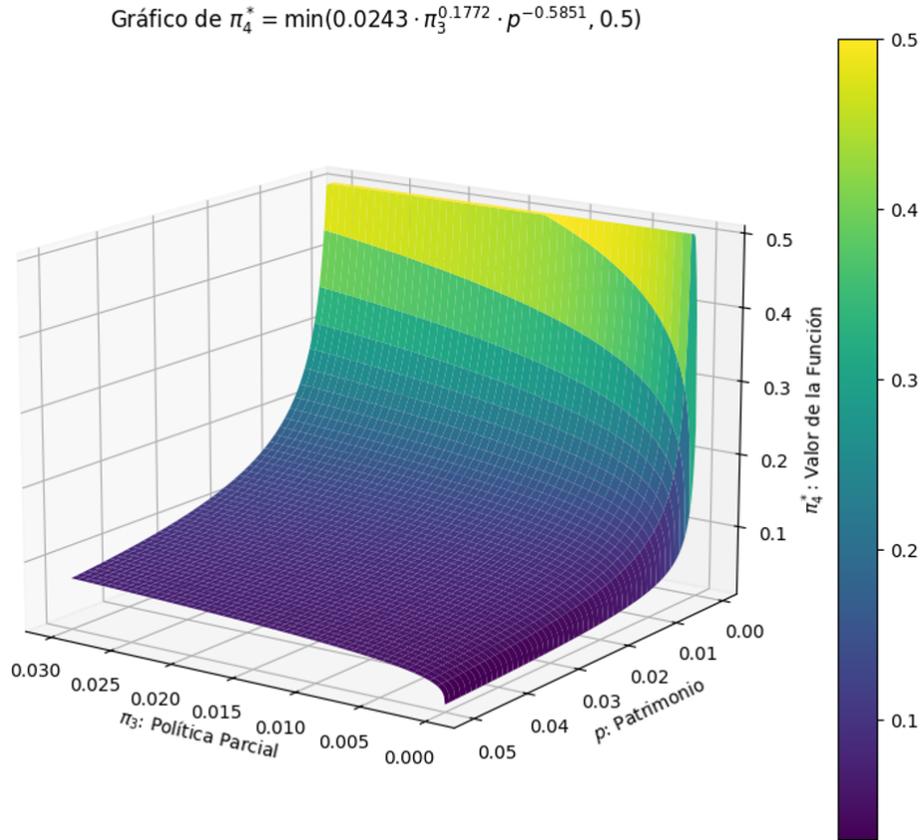


Figura 8: Modelo Compuesto Optimizado para ($H=1000, R=0$)

- Problema de Optimización: ($H = 1000, R = 1$) :

Finalmente, se obtuvieron los valores de los 3 parámetros que definen el modelo de apuesta compuesta para el caso ($H = 1000, R = 1$). El modelo resultó menos agresivo que para el caso ($H = 100, R = 1$), apostando 0,5 (50% del patrimonio) solo si el patrimonio es menor a $\sim 0,95$. Además, para este caso el valor de la política parcial π_3 sí resulta relevante para determinar la apuesta, mientras que el valor del patrimonio p es menos determinante que para el caso de 100 rondas ($H = 100, R = 1$). Esto sugiere que a medida que aumentan las rondas que componen la sesión, el retorno esperado de la baraja de origen cobra mayor relevancia respecto al patrimonio actual del jugador (bajo el contexto de la estructura algebraica que define al modelo compuesto):

$$\pi_4^*(b, p, n; \alpha^*, k^*, c^*) = k^* \cdot \pi_3^{c^*} \cdot p^{\alpha^*} = \min(1,0442 \cdot \pi_3^{0,2734} \cdot p^{-6,1128}, 0,5)$$

A modo de mejorar la visualización de esta estrategia, se muestra la gráfica de las apuestas fraccionarias que realiza, representadas con colores, en función de valores para el patrimonio y para la política parcial π_3 (se consideraron rangos para π_3 y p coherentes con aquellos valores que son más frecuentes durante la sesión de apuestas):

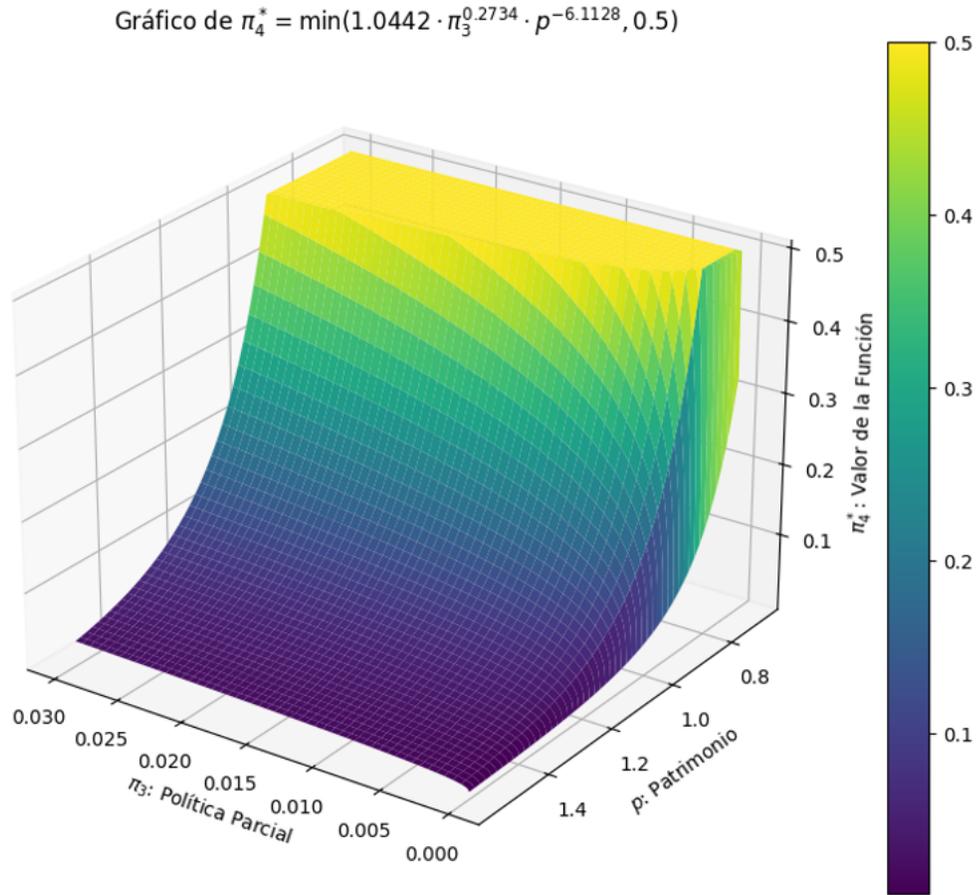


Figura 9: Modelo Compuesto Optimizado para ($H=1000$, $R=1$)

8.3. Distribuciones del Retorno

En esta sección se exponen y analizan las distribuciones del retorno logradas bajo los 4 modelos optimizados durante el proceso de testeo. Los 4 modelos optimizados fueron testados en sesiones de apuestas simuladas a partir de juegos completos filtrados, los cuales fueron seleccionados secuencialmente de manera equiprobable entre la data de testeo. Cada modelo fue aplicado como estrategia de apuestas en 100.000 sesiones de apuestas distintas, alcanzando un determinado “retorno final” en cada una de ellas. A partir de estos 100.000 retornos finales, se generó un histograma y se ajustó la función de densidad de probabilidad (FDP) del retorno asociada al modelo.

A continuación, se presentan los resultados estructurados según los parámetros H y R que definen el problema de optimización, para una mejor visualización. Se aclara que los modelos “Apuesta Porcentual Plana” y “Política Parcial Dependiente de la Baraja” no fueron testeados para los casos (H , $R = 1$), pues como se mostró en la sección previa, los modelos no cambian y los resultados son los mismos que para el caso (H , $R = 0$).

8.3.1. Problema de Optimización: ($H = 100$, $R = 0$)

A partir de la muestra de retornos finales alcanzados, se estimó la FDP para cada modelo a través de la función de densidad de kernel (KDE). El análisis de las FDP obtenidas sugiere una gran similitud entre los resultados esperados bajo el modelo de apuesta porcentual plana y el modelo de política parcial dependiente de la baraja. Sin embargo, la política parcial superó levemente al modelo porcentual plano en cuanto a la mediana de los retornos obtenidos (0,40 % vs 0,33 %), sugiriendo la ventaja de considerar la baraja de origen específica de la ronda (política parcial), en contraste con solo considerar la baraja de origen “promedio” asociada a las barajas filtradas (porcentual plana). El modelo de política parcial también superó al porcentual plano en cuanto al retorno esperado (0,98 % vs 0,56 %), aunque con una mayor desviación estándar asociada (9,96 % vs 8,24 %).

Los modelos de apuesta neta plana y compuesta lograron una mediana significativamente mayor que los otros 2 modelos (1,93 % y 2,09 % respectivamente), demostrando el valor de incorporar el patrimonio actual como variable para la política. Se destaca el hecho de que el modelo de apuesta neta plana igualó al modelo compuesto en cuanto a retorno esperado (1,63 %) y obtuvo una mediana similar, aunque a expensas de una mayor desviación estándar (22,88 % vs 19,80 %). Esto sugiere que bajo la restricción sobre el riesgo de ruina una estrategia de apuesta neta plana resulta ser una buena política en cuanto a rendimiento y simplicidad.

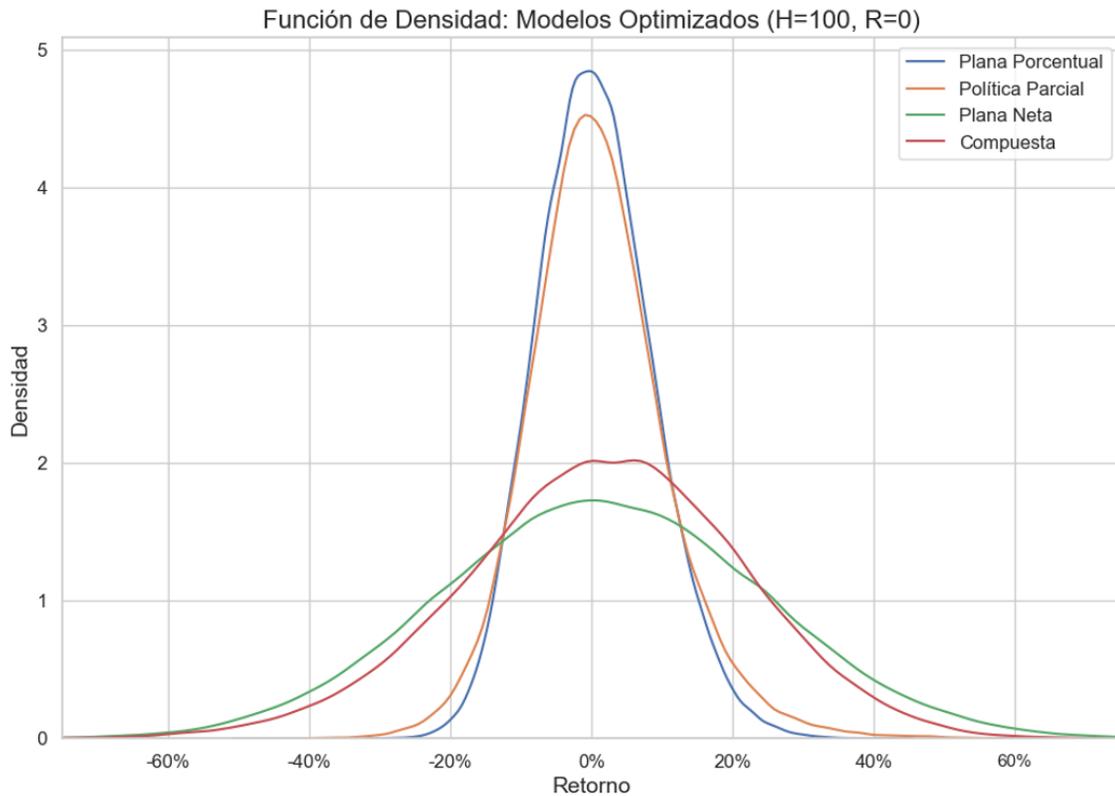


Figura 10: Función de Densidad de Kernel. Modelos Optimizados para (H=100, R=0)

MODELO	Plana Porcentual	Política Parcial	Plana Neta	Compuesta
ROR	0,00 %	0,00 %	0,00 %	0,00 %
MEDIANA	0,33 %	0,40 %	1,93 %	2,09 %
ESPERANZA	0,56 %	0,98 %	1,63 %	1,63 %
DESV. ESTÁNDAR	8,24 %	9,96 %	22,88 %	19,80 %

Tabla 3: Indicadores Estadísticos del Rendimiento de los Modelos (H=100, R=0)

8.3.2. Problema de Optimización: ($H = 100$, $R = 1$)

Debido a la posibilidad de alcanzar la ruina (ROR), la estimación de la FDP a través de la KDE no fue posible, y en cambio se optó por mostrar los histogramas de los retornos finales alcanzados para ambos modelos. Los casos en que se alcanzó la ruina no se consideraron en los histogramas y fueron contabilizados respecto a los 100.000 retornos finales alcanzados, con el fin de calcular el *risk of ruin* (ROR) de la estrategia, el cual se muestra en un recuadro en la gráfica.

El análisis de los histogramas permite concluir que el modelo de apuesta compuesta obtiene una mediana muy superior al modelo de apuesta plana neta (73,95 % vs 0,00 %), aunque a

expensas de más desviación estándar (97,54 % vs 88,93 %) y un riesgo de llegar a la ruina notablemente mayor (47,48 % vs 23,55 %). Mientras que el modelo compuesto presenta una distribución de retornos que concentra el $\sim 52\%$ de los casos en el rango [$\sim 70\%$, $\sim 130\%$] y el $\sim 48\%$ restante en la ruina, el histograma para el modelo de apuesta plana neta sugiere una distribución más uniforme, sobre todo excluyendo valores extremos para el retorno. Las consideraciones recién mencionadas muestran que si bien el modelo compuesto obtuvo un mucho mejor rendimiento respecto al criterio de optimalidad (mediana del retorno), el modelo de apuesta neta plana fue superior en cuanto a riesgo de ruina, volatilidad (desviación estándar), y retorno esperado (5,24 % vs 1,07 %), sugiriendo la necesidad de establecer un criterio de optimalidad más riguroso que el propuesto.

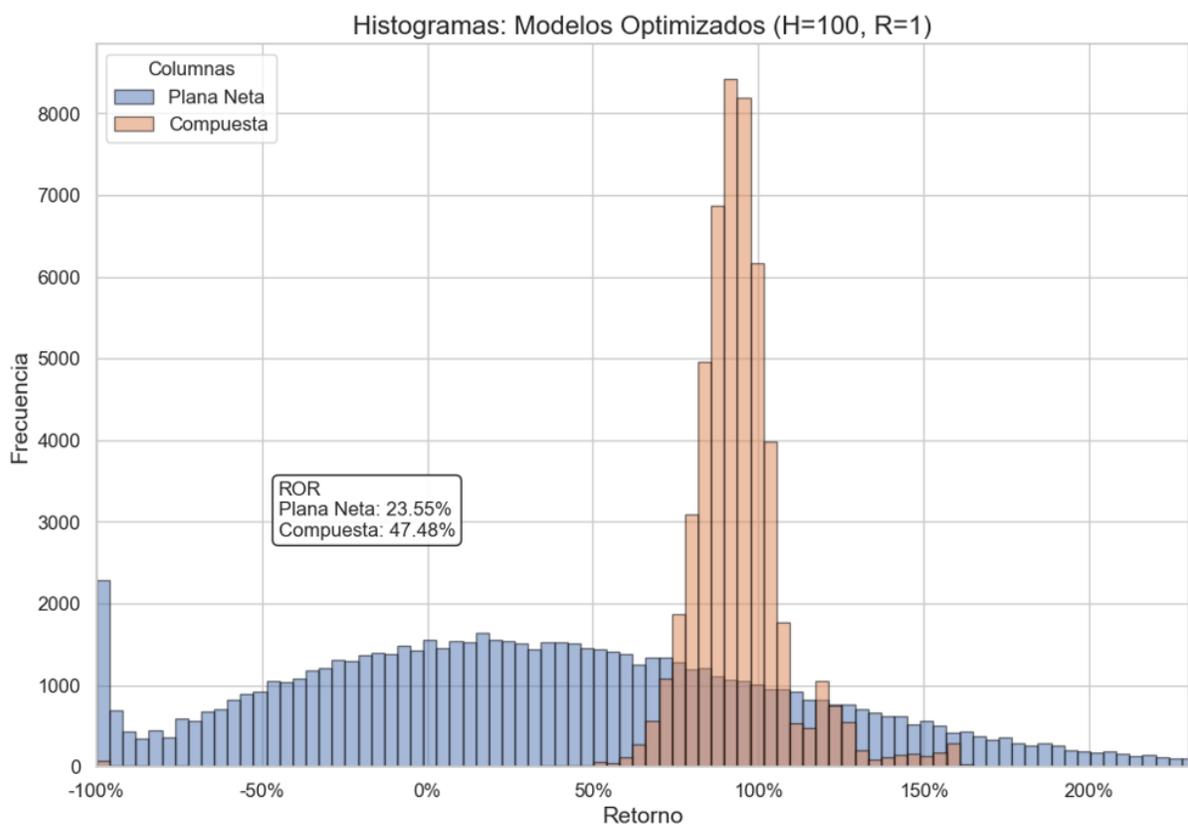


Figura 11: Histograma de Retornos finales. Modelos Optimizados para (H=100, R=1)

	Plana Neta	Compuesta
ROR	23,55 %	47,48 %
MEDIANA	0,00 %	73,95 %
ESPERANZA	5,28 %	1,59 %
DESV. ESTÁNDAR	88,93 %	97,54 %

Tabla 4: Indicadores Estadísticos del Rendimiento de los Modelos (H=100, R=1)

8.3.3. Problema de Optimización: ($H = 1000$, $R = 0$)

A partir de la muestra de retornos finales alcanzados, se estimó la FDP para cada modelo a través de la función de densidad de kernel (KDE). El análisis de las FDA obtenidas sugiere que los 4 modelos optimizados logran resultados similares en cuanto a la forma general de la distribución del retorno. Esto contrasta con el caso ($H = 100$, $R = 0$), donde el modelo compuesto y de apuesta plana presentaron distribuciones marcadamente menos concentradas que los otros 2 modelos, con por lo menos el doble de desviación estándar.

En cuanto a la mediana de los retornos, la ventaja que obtuvo la política parcial respecto al modelo de apuesta porcentual (4,94% vs 2,76%) fue aún mayor que para el caso ($H = 100$, $R = 0$), reforzando la noción de que la ventaja de la política parcial sobre la porcentual se acentúa a medida que la sesión es más larga. Por otro lado, el modelo compuesto mantuvo una mediana levemente superior al modelo de apuesta neta plana (6,93% vs 6,36%), como era esperado, aunque con una desviación estándar considerablemente mayor (36,04% vs 29,95%). Por último, la brecha entre los rendimientos logrados por el modelo de apuesta plana neta y la política parcial se redujo notablemente respecto al caso ($H = 100$, $R = 0$), respaldando el resultado teórico de que en algún momento (para algún valor de H), la política parcial debiera igualar o superar al modelo de apuesta plana neta en cuanto a la mediana de los retornos de la sesión.

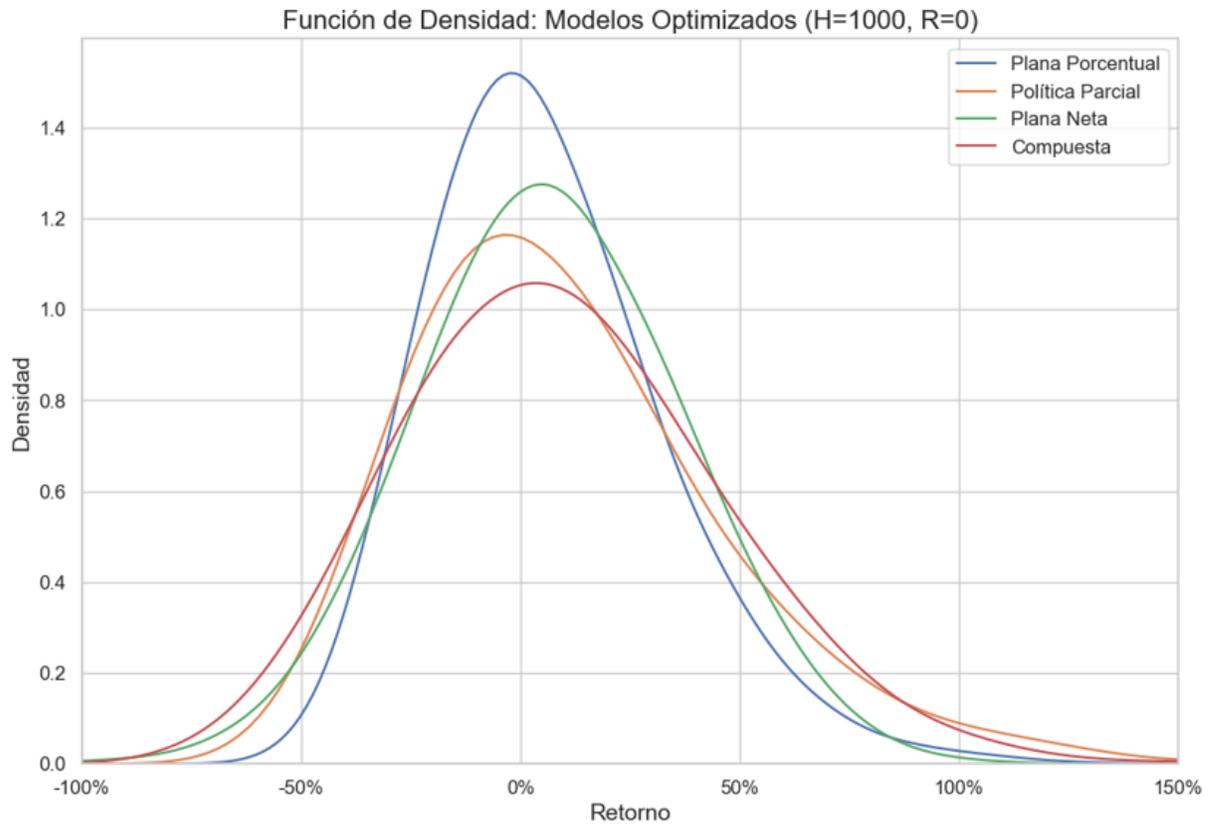


Figura 12: Función de Densidad de Kernel. Modelos Optimizados para (H=1000, R=0)

	Plana Porcentual	Política Parcial	Plana Neta	Compuesta
ROR	0,00 %	0,00 %	0,06 %	0,00 %
MEDIANA	2,76 %	4,94 %	6,36 %	6,93 %
ESPERANZA	6,08 %	10,63 %	6,64 %	9,23 %
DESV. ESTÁNDAR	26,83 %	36,68 %	29,95 %	36,04 %

Tabla 5: Indicadores Estadísticos del Rendimiento de los Modelos (H=1000, R=0)

8.3.4. Problema de Optimización: ($H = 1000$, $R = 1$)

Debido a la posibilidad de alcanzar la ruina (ROR), la estimación de la FDP a través de la KDE no fue posible, y en cambio se optó por mostrar los histogramas de los retornos finales alcanzados para ambos modelos. Los casos en que se alcanzó la ruina no se consideraron en los histogramas y fueron contabilizados respecto a los 100.000 retornos finales alcanzados, con el fin de calcular el *risk of ruin* (ROR) de la estrategia, el cual se muestra en un recuadro en la gráfica.

El análisis de los histogramas obtenidos revela resultados inesperados. Mientras que el modelo de apuesta plana neta mejoró considerablemente su rendimiento respecto al caso ($H = 100, R = 1$), obteniendo una mediana mucho mayor (19,67% vs 0,00%), el modelo compuesto obtuvo un rendimiento levemente menor (73,23% vs 73,95%). En cuanto al retorno esperado, el modelo de apuesta plana obtuvo un 20,79%, casi 4 veces mayor al caso ($H = 100, R = 1$), manteniendo un riesgo de ruina (ROR) similar (23,36% vs 23,55%). De la misma forma, el modelo compuesto tampoco varió significativamente el riesgo de ruina (47,17% vs 47,85%), y obtuvo un retorno esperado casi 6 veces mayor (9,00% vs 1,59%) que para el caso ($H = 100, R = 1$). De aquí se extrae que, si bien el modelo compuesto no mejoró la mediana obtenida al aumentar la cantidad de rondas de la sesión de 100 a 1000 como era esperado, sí mejoró sus rendimientos en general, obteniendo un mayor retorno esperado y manteniendo constante el ROR.

Los resultados confirman la observación realizada para otros casos sobre las limitaciones de la metodología basada en modelos algebraicos para la política, y refuerzan la necesidad de establecer un criterio de optimalidad más riguroso que el propuesto.

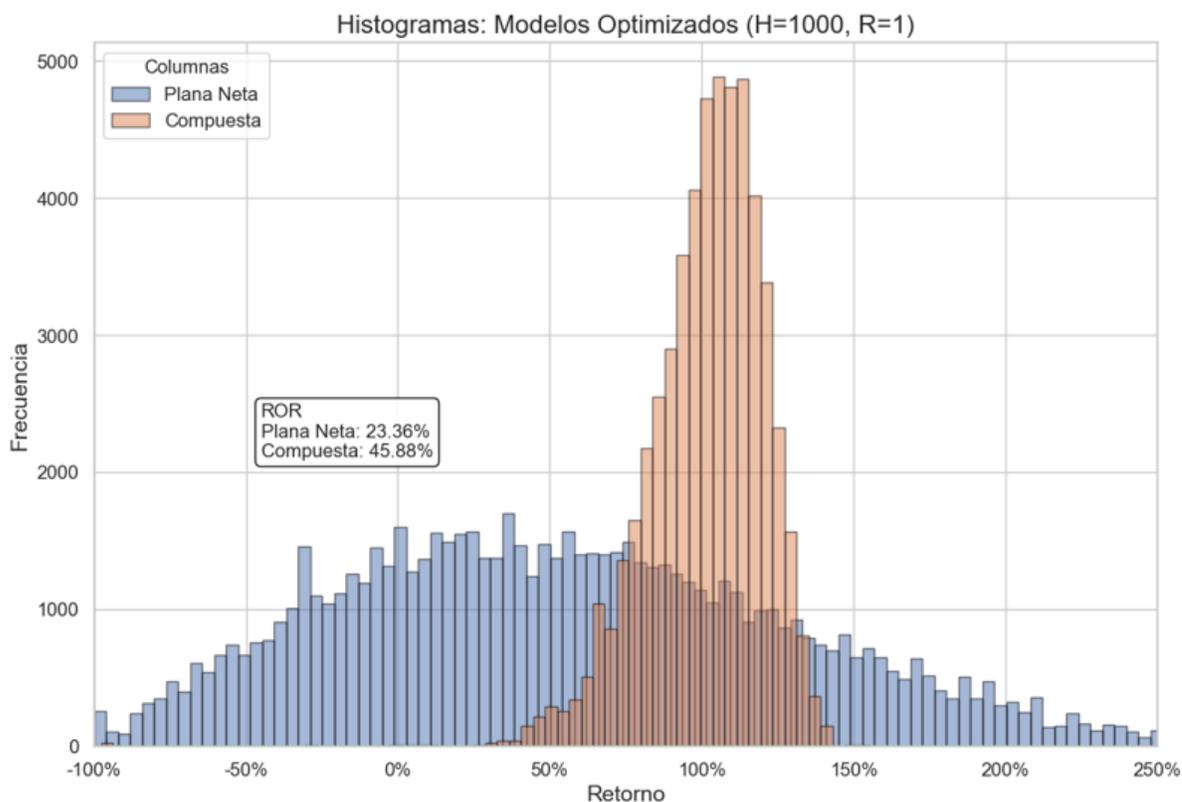


Figura 13: Histograma de Retornos finales. Modelos Optimizados para ($H=1000, R=1$)

	Plana Neta	Compuesta
ROR	23,36 %	45,88 %
MEDIANA	19,67 %	73,23 %
ESPERANZA	20,79 %	9,00 %
DESV. ESTÁNDAR	95,06 %	101,35 %

Tabla 6: Indicadores Estadísticos del Rendimiento de los Modelos (H=1000, R=1)

9. Discusión

El presente estudio ha explorado las dos estrategias fundamentales que determinan completamente el comportamiento de un jugador de blackjack: la política de juego y la política de apuestas. Aunque la política de juego óptima fue previamente establecida en "Las matemáticas del blackjack" de Ricardo E. Peytaví, la metodología propuesta en este trabajo ha evidenciado ser vastamente superior en cuanto a eficiencia computacional. En contraste con la demora de varios minutos que requiere el enfoque recursivo del trabajo mencionado para calcular decisiones óptimas, el algoritmo desarrollado en esta investigación logra el mismo objetivo en menos de un segundo. Esta notable mejora en la eficiencia computacional ha facilitado la deducción de la Función de Masa de Probabilidad del retorno de una ronda antes de su inicio, un aspecto no abordado en el trabajo de Peytaví, y que resulta fundamental para estudiar las políticas de apuesta.

Los hallazgos de este estudio corroboran la idea, ampliamente sostenida y respaldada por investigaciones anteriores [5], de que el retorno esperado por ronda en el blackjack no es fijo, sino que fluctúa en función de la composición de la baraja. En el marco de las reglas y condiciones especificadas para este estudio, se constató que el jugador obtiene una ventaja sobre la casa en aproximadamente el 20% de las rondas. Este dato resalta la relevancia de adoptar estrategias de apuesta que se ajusten dinámicamente al conocimiento de la composición de la baraja, planteando además la cuestión de cómo esta ventaja podría variar bajo diferentes configuraciones de reglas, y cuales son las mas ventajosas para el jugador.

Respecto al problema de determinar la política de apuestas óptimas, se destaca haber modelado completamente el problema como Proceso de Decisión de Markov, especificando explícitamente la función de transición y una función de recompensa coherente con principios económicos, basada en una función de utilidad tipo CRRA. Este enfoque teórico, en combinación con el programa desarrollado, constituyen los elementos esenciales para resolver completamente el problema de optimización para la política de apuestas mediante el uso de redes neuronales de refuerzo. Un aspecto fundamental que debe ser estudiado con mayor profundidad en la implementación de dicha red, consiste en la exploración de otras funciones de utilidad sobre los retornos, como las funciones CARA, HARA, o funciones diseñadas específicamente. La forma de la función de utilidad determina la política de apuestas, y por lo tanto impacta directamente en la distribución de los retornos asociados a una sesión de apuestas. Es esencial investigar la relación entre la función de utilidad que se establece y las distribuciones del retorno asociadas a una sesión de apuestas.

Un caso particularmente interesante del problema de optimización para la política de apuestas corresponde al caso de política parcial, la cual considera únicamente la composición de la baraja como variable, sin tomar en cuenta el patrimonio actual y la cantidad de rondas jugadas. Se destaca haber desarrollado matemáticamente el problema para la política parcial óptima (bajo el supuesto de que se juegan infinitas rondas) bajo una función de utilidad tipo

CRRA, culminando con una ecuación fácilmente resoluble (mediante métodos computacionales) para hallar la solución. Además, parte valiosa del desarrollo teórico consiste en haber encontrado explícitamente una aproximación para la distribución del retorno bajo dicha política, que resulta ser una distribución log normal. La posibilidad de resolver una versión simplificada del problema de manera precisa, en conjunto con los resultados matemáticos logrados, constituyen una buena exploración teórica y práctica para la resolución general del problema de política de apuestas. Finalmente, el mecanismo desarrollado para hallar la política parcial óptima puede ser directamente aplicable a cualquier problema de gestión de trading donde se conozca la distribución del retorno por periodo. Se destaca la relevancia de este aspecto, pues permite estudiar fácilmente las propiedades de la gestión de trading eficiente para diversas configuraciones de distribución del retorno por periodo.

Respecto a los resultados obtenidos en el experimento, los rendimientos alcanzados por los modelos optimizados se consideran prometedores, y superan el rendimiento de cualquier estrategia de apuestas para el Blackjack publicada hasta la fecha. Es particularmente destacable la diversidad en el desempeño de los distintos modelos. Por ejemplo, el modelo compuesto logró alcanzar una mediana del retorno de hasta el 73 %, manteniendo un retorno esperado del 2 %, en apenas 100 rondas. Por otro lado, en un horizonte de 1000 rondas, el modelo plano neto para la política logró un retorno esperado y una mediana del retorno de 20 %. Esto sugiere una gran flexibilidad para la distribución del retorno de una sesión, según la estrategia que se defina. Un aspecto especialmente interesante fue la comparación del rendimiento de la política parcial con modelos que incorporan además el patrimonio actual como variable, demostrando estos últimos rendimientos muy superiores, a pesar de ser modelos algebraicamente simples. Estos resultados son un buen comienzo como exploración de la solución general, y sugieren un gran potencial de rendimiento para la política de apuesta óptima generalizada. Sin embargo, el análisis de los modelos optimizados también reveló una deficiencia en la mediana del retorno como función objetivo, por lo que la solución general debe abordar otras funciones objetivo más robustas. Una buena métrica de rendimientos podría ser el Sharpe Ratio de los retornos, aunque conlleva dificultades técnicas asociada a la implementación de la red neuronal, al no contar con propiedades de linealidad que la hagan fácilmente traducible a una función de recompensa inmediata.

Un último punto de discusión es el valor de esta investigación en un contexto mas amplio. Se hace notar con énfasis que el planteamiento matemático para hallar la política de apuestas óptima establecido en este trabajo es directamente aplicable al problema de gestión de portafolio. Solucionar el problema general para el Blackjack es equivalente con encontrar la estrategia de trading óptima para un activo que presentase retornos por periodo con igual FMP que los del Blackjack. Sin embargo, los activos financieros presentan un retorno por periodo que distribuye de manera continua (en el Blackjack es discreto), y su función de densidad solo puede ser estimada, haciéndolo un problema inherentemente más complejo. Aun así, es posible aproximar distribuciones continuas por medio de distribuciones discretas y

viceversa, haciendo del Blackjack un caso de estudio ideal para la comprensión del problema general del portafolio. Sin duda la solución general del Blackjack conllevará aprendizajes y resultados directamente aplicables a la optimización de la gestión de trading.

10. Conclusión

En el presente trabajo se modelaron matemáticamente de manera rigurosa los 2 problemas que debe resolver el jugador de Blackjack: establecer una política de juego y establecer una política de apuestas. Computar la política de juego óptima se considera resuelto en la comunidad de manera extraoficial, pero sin un marco teórico riguroso. En esta tesis se concluyó con un programa original que computa la política de juego óptima y la FMP del retorno por ronda en tiempo real, aproximando el caso asociado a “Abrir” la mano. Este programa permitió estudiar la rentabilidad para el jugador de Blackjack bajo política de juego óptima según las reglas establecidas, obteniendo un retorno esperado general negativo de $-0,63\%$. Sin embargo, también se encontró que en aproximadamente un 20% de las rondas el retorno esperado para el jugador es positivo, promediando un retorno esperado de $0,82\%$, reafirmando la idea de que es posible obtener ventaja sobre la casa escogiendo en que rondas participar.

En cuanto a la política de apuestas, este trabajo presenta una descripción precisa del problema de optimización como un Proceso de Decisión de Markov, incorporando explícitamente la función de transición y una función de recompensa asociada a una función de utilidad tipo CRRA. Se desarrolló teóricamente una versión simplificada del problema, que asume sesiones de apuestas infinitas y una política de apuestas parcial que solo considera la composición de la baraja como variable para determinar la apuesta. El desarrollo culminó con una ecuación como condición suficiente para calcular la política parcial óptima, y una aproximación de la distribución de retornos asociada a dicha política. El testeo de la política parcial óptima en conjunto con sencillos modelos algebraicos propuestos para la exploración de soluciones concluyó con los mejores rendimientos publicados hasta la fecha para un jugador de Blackjack, alcanzado una mediana del retorno de hasta el 73% en 100 rondas, y un retorno esperado de 20% en 1000 rondas.

El desarrollo de esta investigación representa un avance significativo en la solución del problema del apostador de Blackjack, así como en el ámbito de la optimización de la gestión de portafolios. El modelamiento matemático propuesto en este estudio es directamente aplicable a la gestión de trading, lo que sugiere que la solución general del Blackjack podría resultar de gran utilidad para la resolución del problema de optimización de portafolios.

11. Bibliografía

- [1] Baldwin, R., Cantey, W., Maisel, H., & McDermott, J. (1956). *The Optimum Strategy in Blackjack*. Recuperado de: https://web.williams.edu/Mathematics/sjmiller/public_html/341Fa09/handouts/baldwin_OptimalStrategyBlackjack.pdf
- [2] Tejada, J. L. (1985). *Estudio de la estrategia óptima para el blackjack mediante simulaciones de mazos estáticos*. Universidad Complutense de Madrid. Recuperado de: <https://eprints.ucm.es/id/eprint/16268/1/Tejada18.pdf>
- [3] Vince, R., & Zhu, Q. (2013). *Optimal Betting Sizes for the Game of Blackjack*. Advanced Risk & Portfolio Management® Research Paper Series. <https://doi.org/10.2139/ssrn.2324852>
- [4] Werthamer, N. R. (2005). *Optimal Betting in Casino Blackjack*. International Gambling Studies, 5(2), 253-270. <https://doi.org/10.1080/14459790500303485>
- [5] Kamron Jensen. (2014). *The Expected Value of an Advantage Blackjack player*. Recuperado de: <https://digitalcommons.usu.edu/cgi/viewcontent.cgi?article=1528&context=gradreports>
- [6] Kelly, J. L. (1956). *A New Interpretation of Information Rate*. Bell System Technical Journal. Recuperado de: https://www.princeton.edu/~wbialek/rome/refs/kelly_56.pdf
- [7] Peytaví Machado, R. E. (2018). *Las Matemáticas del Blackjack*. Universidad de La Laguna. Recuperado de: <https://riull.ull.es/xmlui/bitstream/handle/915/18205/TFG%20Ricardo%20E%20Peytav%C3%AD%20Machado.pdf?sequence=4&isAllowed=y>
- [8] Alias “MGP”. *Free Blackjack Combinatorial Analyzer*. Recuperado de: <https://www.blackjackinfo.com/free-blackjack-combinatorial-analyzer/>
- [9] Alias “k_c”. *Composition Dependent Combinatorial Analyzer (cdca)*. Recuperado de: <https://www.bjstrat.net/cgi-bin/cdca.cgi>
- [10] Wikipedia. (2024). *Conteo de cartas*. Recuperado de: https://es.wikipedia.org/wiki/Conteo_de_cartas

12. Anexos

Anexo A. Pseudocódigo Ilustrativo de *CrupierProb*

```
1 def CrupierProb(w_c=(C, z, v, q)):  
2  
3     #Se inicializa d como un vector de 22 elementos todos igual a 0.  
4     # d[0] representa la probabilidad de Blackjack  
5     # d[16], d[17],...,d[20] representan las probabilidades de finalizar  
6     con una suma de 17, 18, 19, 20, 21 respectivamente.  
7     # d[21] representa la probabilidad de pasarse  
8  
9     d=[0, 0, 0, ..., 0]  
10  
11     #Si w_c es una instancia final de la ronda finaliza la recursion.  
12     if (20 >= C >= 17):  
13         d[C-1]=1  
14  
15     if (C > 21 and z==0):  
16         d[21]=1  
17  
18     if C==21:  
19         if v==2:  
20             d[0]=1  
21         else:  
22             d[20]=1  
23  
24     else:  
25         #Se reparte una carta de la baraja q al crupier.  
26         for carta in range(1, 11):  
27             #Se revisa si quedan cartas en la baraja  
28             if q[carta-1] > 0:  
29  
30                 #Se actualizan las variables de instancia segun carta.  
31                 w_c = (C, z, v, q)  
32  
33                 #Se calcula la probabilidad de ocurrencia de la carta.  
34                 Prob_Carta= Prob(carta, q)=q[carta-1]/sum(q)  
35  
36                 #Se actualiza el vector d recursivamente.  
37                 d = d + CrupierProb(w_c)*Prob_Carta  
38  
39     return d
```

Anexo B. Pseudocódigo Ilustrativo de *JugadorDecide*

```
1 def JugadorDecide(w_j=(J^1, x^1, y^1, C, z, q)):  
2  
3     #Se calcula CrupierProb para la baraja q.
```

```

4     M = CrupierProb(C, z, 1, q)
5
6     #gpS=[prob ganar si se Planta, prob perder si se Planta]
7     #gpH=[prob ganar si se Pide, prob perder si Pide]
8
9     gpS = Auxiliar(J^1, x^1, y^1, M)
10    gpH = [0, 0]
11
12    #Si w_j es una instancia terminal para el jugador finaliza la
13    #recursion.
14    if J^1 >= 21:
15        #En las instancias terminales el jugador siempre pierde si
16        #pide.
17        gpH = [0, 1]
18
19    else:
20        #Se reparte una carta de la baraja q al jugador.
21        for carta in range(1, 11):
22            #Se revisa si quedan cartas en la baraja
23            if q[carta-1] > 0:
24
25                #Se actualizan las variables de instancia segun
26                #carta.
27                w_j=( J^1, x^1, y^1, C, z, q)
28
29                #Se calcula la probabilidad de ocurrencia de la
30                #carta.
31                Prob_Carta= Prob(carta, q)= q[carta-1]/sum(q)
32
33                #Se actualiza el vector gpH segun la funcion de
34                #transicion.
35                gpH[0]= gpH[0] + JugadorDecide(w_j)[1]*Prob_Carta
36                gpH[1]= gpH[1] + JugadorDecide(w_j)[2]*Prob_Carta
37
38    esperanzaS = gpS[0] - gpS[1]
39    esperanzaH = gpH[0] - gpH[1]
40
41    #Se determina la mejor decision a partir de la esperanza.
42    if esperanzaS >= esperanzaH:
43        decision = [0, gpS[0], gpS[1], esperanzaS]
44
45    else:
46        decision = [1, gpH[0], gpH[1], esperanzaH]
47
48    return decision

```

Anexo C. Pseudocódigo Ilustrativo de *ProbSecsCrup*

```

1 def ProbSecsCrup(C, q):
2
3     #Matriz_Total=[ci, cf, sec_crup, mult]: Contiene todas las
4     #posibles Secuencias unicas de cartas que puede recibir el crupier
5     #para cada uno de los 10 posibles ci.
6
7     #ci: Puntaje inicial del crupier. Puede ser 0, o desde el 2 hasta
8     #el 11.
9
10    #cf: representa el puntaje final del crupier. -1 si Bj. 22 si se
11    #pasa.
12
13    #sec_crup: vector de 10 componentes. Representa las cartas que
14    #conforman la mano del crupier al final de la ronda, incluyendo ci
15
16    #mult: representa la multiplicidad de sec_crup excluyendo ci.
17
18    #Se usa Matriz_Total, precalculada.
19    TOTAL=Matriz_Total
20
21    #Se genera una copia de TOTAL, para modificarla.
22    Matriz_Probs=TOTAL.copy()
23
24    for fila in Matriz_Probs:
25        #C==0: El crupier no ha recibido su primera carta.
26        if C==0:
27            #Se calcula la probabilidad de que ocurra la secuencia desde
28            #la baraja q, sin considerar combinaciones.
29            P_s=Prob(sec_crup, q)
30
31            #C!=0: El crupier ya recibio su primera carta.
32            elif C!=0:
33                #Se saca la carta asociada a ci de la secuencia.
34                sec_crup = sec_crup - carta_ci
35
36                #Analogo a la rama C==0
37                P_s=Prob(sec_crup, q)
38
39            #Se calcula la probabilidad total de la secuencia, considerando su
40            #multiplicidad.
41            P_s= P_s*mult
42
43            #Se reemplaza el valor de la multiplicidad por P_s, dado que ya se uso
44            #.
45            mult=P_s
46
47            #Se retorna la matriz resultante, que tiene la
48            #forma Matriz_Probs=[ci, cf, sec_crup, P_s]

```

```
48 return Matriz_Probs
```

Anexo D. Pseudocódigo Ilustrativo de *GanarPerder*

```
1 def GanarPerder(J^1, x^1, y^1, C, Matriz_Probs):
2
3     #Se trabaja con Matriz_Probs=[ci, cf, sec_crup, P_s]
4
5     #Se incializa el vector gpS vacio.
6     gpS=[0, 0]
7
8     #Si C es distinto de 0, solo se trabaja con las filas en que ci==C
9     if C!=0:
10        Matriz_Probs = Matriz_Probs[Matriz_Probs[:, 0] == C]
11
12    for fila in Matriz_Probs:
13        #Se ve el pago a partir del puntaje final del crupier cf.
14        #Puede ser 1 si gana el jugador o -1 si pierde.
15        pago= retorno(J^1, x^1, y^1, cf)
16
17        if pago==1:
18            gpS[0]= gpS[0]+ P_s
19
20        elif pago==-1:
21            gpS[1]= gpS[1]+ P_s
22
23    return gpS
```

Anexo E. Pseudocódigo Ilustrativo de *Bayes*

```
1 def Bayes(Matriz_Probs, carta, C, y^1, q):
2
3     #Matriz_Probs =[ci, cf, sec_crup, P_s]: Calculada para una
4     #determinada baraja de ronda q cualquiera.
5
6     #Carta: es la carta extraida de la baraja q.
7
8     #Se actualiza cada P_s de la matriz, usando teorema de Bayes.
9     for fila de Matriz_Probs:
10        #Caso especial: La carta va a la mano del crupier.
11        if C==0 and y^1==1:
12            #Se saca la carta de la secuencia.
13            sec_crup= sec_crup - carta
14
15            #Se actualiza la probabilidad de las nuevas secuencias.
16            P_s= P_s/Prob(carta, q)
17
```

```

18     #Caso normal: La carta va a la mano del jugador.
19     else:
20         P_s= P_s*Prob(carta, (q-sec_crup))/Prob(carta, q)
21
22     return Matriz_Probs

```

Anexo F. Pseudocódigo Ilustrativo de *JugadorDecideRapido*

```

1 def JugadorDecideRapido(w_j=(J^1, x^1, y^1, C, z, q), Matriz_Probs==None):
2
3     #En la primera llamada se inicializa Matriz_Probs, segun C y q.
4     if Matriz_Probs==None:
5         Matriz_Probs= ProbSecsCrup(w_c=(C, q))
6
7     #Se calcula el vector gpS.
8     gpS = GanarPerder(J^1, x^1, y^1, C, Matriz_Probs)
9     gpH = [0, 0]
10
11     #Si w_j es una instancia terminal para el jugador finaliza la
12     #recursion.
13     if J^1 >= 21:
14         #En las instancias terminales el jugador siempre
15         #pierde si pide.
16         gpH = [0, 1]
17
18     else:
19         #Se reparte una carta de la baraja q al jugador.
20         for carta in range(1, 11):
21             #Se revisa si quedan cartas en la baraja.
22             if q[carta-1]>0:
23                 #Se actualizan las variables de instancia segun carta.
24                 w_j=(J^1, x^1, y^1, C, z, q)
25
26                 #Se calcula la probabilidad de ocurrencia de la carta.
27                 Prob_Carta= Prob(carta, q)=q[carta-1]/sum(q)
28
29                 #Se actualiza Matriz_probs segun carta.
30                 Matriz_probs= Bayes(Matriz_Probs, carta, C, y^1, q)
31
32                 #Se actualiza el vector gpH segun
33                 #la decision y Prob_Carta.
34                 res= JugadorDecideRapido(w_j,Matriz_Probs)
35
36                 gpH[0]= gpH[0] + res[1]*Prob_Carta
37                 gpH[1]= gpH[1] + res[2]*Prob_Carta
38

```

```

39     esperanzaS = gpS[0] - gpS[1]
40     esperanzaH = gpH[0] - gpH[1]
41
42     #Se determina la mejor decision a partir de la esperanza.
43     if esperanzaS >= esperanzaH:
44         decision = [0, gpS[0], gpS[1], esperanzaS]
45
46     else:
47         decision = [1, gpH[0], gpH[1], esperanzaH]
48
49     return decision

```

Anexo G. Pseudocódigo Ilustrativo de *JugadorAbreExacta*

```

1 def JugadorAbreExacta(J^1, C, q):
2
3     #Matriz_Jugador= [J^1, secj_1, secj_2, m_1, m_2]: Contiene
4     #en todas las combinaciones de secuencias que puede recibir el
5     #jugador en ambas manos, que comienzan por una unica carta
6     #asociada a un puntaje J^1.
7
8     # J^1: Puntaje de cada una de las 2 manos del
9     #crupier, igual para ambas.
10
11     #secj_i: Secuencia de cartas que recibe la mano i del
12     #jugador. Estas secuencias se deducen de la politica optima
13     #computada por J.D.Rapido.
14
15     #m_i: multiplicidades de las secuencias, tambien rescatadas
16     #del computo de J.D.Rapido.
17
18     #Se usa Matriz_Jugador, construida tras ejecutar J.D.Rapido.
19     Matriz_J= Matriz_Jugador
20
21     #Se inicializa Matriz_Probs, segun C y q.
22     Matriz_Probs= ProbSecsCrup(C, q)
23
24     #gpH= prob. de [ganar ambas, perder ambas, ganar una
25     #y empatar otra, perder una y empatar otra]
26     gpA = [0, 0, 0, 0]
27
28     for fila in Matriz:
29         for fila in Matriz:
30             #Se calcula la secuencia de cartas total que extrae
31             #el jugador con ambas manos.
32             SecTot= secj_1 + secj_2

```

```

33
34     #Se deduce el puntaje de cada mano, considerando
35     #las secj_i.
36     {J_prime}^1= J^1 + secj_1
37     {J_prime}^2= J^1 + secj_2
38
39     #Se calcula la probabilidad de ocurrencia de SecTot.
40     Prob_SecTot= Prob(SecTot, q)*m_1*m_2
41
42     #Se deduce Matriz_Res segun SecTot y Matriz_Probs.
43     Matriz_Res= Bayes_aux(Matriz_Probs, SecTot, q)
44
45     #Se calcula el vector gpA.
46     gpA = GanarPerder_aux({J_prime}^1, {J_prime}^2, C,
47     Matriz_Res)
48
49     #Se pondera cada uno de los 4 posibles casos gpA[i]
50     #por su prob, actualizando gpH.
51     for i in range(4):
52         gpA[i]= gpA[i] + gpA[i]*Prob_SecTot
53
54     EspAbre = 2*(gpA[0]-gpA[1]) + (gpA[2]-gpA[3])
55
56     return EspAbre

```