



Universidad
Carlos III de Madrid
www.uc3m.es



Universidad
Carlos III de Madrid
www.uc3m.es

TRABAJO FIN DE GRADO

**Título: Diseño y evaluación de una
heurística de juego de póker**

Autor: Jacobo Conrado Pérez-Fajardo

**Titulación: Grado en Ingeniería
Informática**

**Profesor: Antonio Berlanga De Jesús,
José Luís Guerrero Madrid**

Fecha: 12/06/2012



Índice

1.	Introducción.....	6
1.1	Estado del arte.....	8
1.2	Objetivos.....	9
1.3	Introducción al juego del póker.....	11
1.4	Aprendizaje automático vs robots pre programados.....	13
2.	Descripción del dominio.....	15
2.1	Introducción.....	15
2.2	Juego del póker.....	16
2.3	Reglas del póker.....	17
2.3.1	Combinaciones.....	18
2.4	Variante Texas Hold'em.....	22
2.5	Variante con límite fijo.....	23
2.6	Posiciones.....	24
2.7	Fases.....	26
2.7.1	Preflop.....	27
2.7.2	Flop.....	39
2.7.3	Turn.....	40
2.7.4	River.....	41
2.8	Apuestas.....	42
2.9	Combinatoria.....	44
2.10	Póker y aprendizaje automático.....	45
3.	Algoritmos evolutivos.....	46
3.1	Introducción.....	46
3.2	Algoritmos Genéticos.....	47
3.3	Modificaciones al CGA.....	53
3.3.1	Codificación.....	54
3.3.2	Sistema de evaluación co-evolutivo.....	57
3.3.3	Conjunto de entrenadores.....	57
3.3.4	Función de evaluación.....	58
3.3.5	Operadores.....	60
4.	Implementación.....	67
4.1	Introducción.....	67
4.2	Simulador.....	68



4.3	Definición de mano.....	69
4.4	Generador de manos	70
4.5	Optimizaciones	71
4.5.1	Modelo de concurrencia	71
4.5.2	Manos pre-generadas.....	71
4.5.3	Indexación por factores primos	72
4.6	Coste del uso del cromosoma	73
4.7	Criterios de convergencia	75
4.8	Generación de log para PokerTracker 3	76
5.	Resultados.....	77
5.1	Parámetros utilizados.....	78
5.2	Prueba de generación de entrenadores.....	80
5.3	Prueba de generación de individuos contra entrenadores.....	80
5.4	Análisis de resultados	81
5.4.1	Entrenadores generados.....	82
5.4.2	Entrenador 0	84
5.4.3	Entrenador 1	90
5.4.4	Entrenador 2	96
5.4.5	Entrenador 3	102
5.4.6	Entrenador 4	108
5.4.7	Entrenador 5	114
5.4.8	Entrenador 6	120
5.4.9	Entrenador 7	126
5.4.10	Entrenador 8	132
5.4.11	Individuos contra entrenadores.....	138
5.5	Estudio del jugador final.....	144
6.	Presupuesto.....	146
6.1	Planificación	146
6.2	Materiales.....	147
6.2.1	Hardware	147
6.2.2	Software.....	147
6.3	Recursos humanos	147
6.3.1	Diseño.....	148
6.3.2	Implementación	148



6.3.3	Pruebas	149
6.3.4	Documentación.....	149
6.3.5	Coste de personal.....	149
6.3.6	Total.....	150
7.	Conclusiones y trabajos futuros.....	151
8.	Bibliografía.....	153



Agradecimientos

Mis más sinceros agradecimientos a:

Mi tutor Antonio Berlanga y a mi co-tutor José Luis Guerrero por haber estado tan atentos a mi trabajo y por haberme dado todo su apoyo y depositado su confianza en mí.

A mis profesores por todos los conocimientos que me han proporcionado, tanto en el aspecto académico, como en el personal.

A mis padres y a mi hermana por el apoyo y el ánimo que me han dado a lo largo de toda la carrera.

A mis compañeros de clase por haber mostrado tanta ilusión y animo.

A mis amigos que han tenido que oírme hablar del proyecto, póker y algoritmos genéticos durante tantas horas.



1. Introducción

El póker es un juego de cartas con una historia que se remonta varios siglos atrás. Este juego de apuestas conforme se han sucedido los años ha cobrado cada vez más importancia, siendo ahora uno de los juegos de cartas más jugados a nivel mundial.

Los algoritmos genéticos (Goldberg 1989), una técnica de aprendizaje automático creada en la década de los sesenta, han supuesto un gran avance en la rama del conocimiento de la inteligencia artificial. Esta técnica se ha utilizado y se aplica en diversos campos como método para obtener soluciones a problemas. Este método tiene múltiples aplicaciones prácticas (L. Haupt, Ellen Haupt 2004.), desde casos concretos como la predicción de bancarrotas bancarias (Shin y Lee 2002), hasta modelos de aplicación más generales como el aprendizaje de sistemas completos de clasificación (Lanzi 2009).

La aplicación de los algoritmos genéticos puede ser viable para la generación de un jugador de póker automático, debido a que el problema (como se explicará más adelante) cumple con los requisitos para ser tratado en esta técnica, ya que se puede tratar como un problema de optimización. Este proyecto analizará hasta qué punto el uso de algoritmos genéticos puede resolver el problema.

En el presente trabajo se expondrá de forma detallada la aplicación de algoritmos genéticos como método de aprendizaje, y así, obtener estrategias competitivas para el juego del póker. Uno de los mayores retos que supone este proyecto, es intentar alcanzar un jugador con una estrategia de juego razonable, sin introducir conocimiento de dominio, y ver hasta dónde puede llegarse con este sistema.

Se comenzará explicando los objetivos de este trabajo de fin de grado, se abordarán también las diferencias entre la obtención de un jugador de póker con técnicas de aprendizaje automático frente a uno pre programado.

A continuación se explicará con detalle en qué consiste el juego del póker, explicando las reglas de juego y los factores que se han de tener en cuenta si se quiere obtener una estrategia ganadora. En esta sección adicionalmente se especificará para qué variantes del póker se ha creado una solución. Finalmente hablaremos de cuáles son los puntos clave a la hora de aplicar técnicas de aprendizaje automático, para obtener un jugador que rentabilice el juego.

Se continuará explicando la técnica de aprendizaje automático utilizada para cumplir con los objetivos propuestos. Se detallará en qué consiste un algoritmo genético, y cuáles son las bases de su funcionamiento. También se expondrán las modificaciones aplicadas al algoritmo para que se adapte a las necesidades del trabajo. Se mostrarán las diferentes alternativas que se han probado a la hora de implementar la solución y cuál se ha elegido. Adicionalmente se hablará de las diferentes alternativas propuestas para la función de evaluación y cuál se ha elegido finalmente.



En la siguiente sección se explicará cuál ha sido la implementación final del sistema, cuáles han sido los conocimientos teóricos y prácticos obtenidos a lo largo del grado que este proyecto concluirá y que se han utilizado para realizar el trabajo. En esta sección también se explicarán las optimizaciones realizadas para mejorar el rendimiento y obtener mejores tiempos de ejecución.

Adicionalmente se detallarán las pruebas que se han realizado para probar el funcionamiento del sistema desarrollado y qué resultados se han obtenido. Se realizarán diferentes juegos de pruebas, cada uno con un objetivo diferente.

En el apartado siguiente se hará un análisis del presupuesto requerido para realizar el trabajo. Se analizará el coste de las herramientas utilizadas tanto software como hardware, así como, el coste del personal involucrado en el desarrollo del proyecto.

Finalmente se expondrán las conclusiones obtenidas a lo largo y final del proyecto. Por último se abordarán las ideas que se plantean para proyectos futuros y de forma somera, se explicarán en qué consisten y su alcance.



1.1 Estado del arte

En esta sección se repasarán algunas de las técnicas para la generación de jugadores de póker automático.

Las técnicas que se describen a continuación han resultado ser muy exitosas, siendo cada una de ellas mejor que la anterior (en orden de aparición). Esto resulta un buen incentivo para tratar este problema ya que hay precedentes de utilización de técnicas que han tenido éxito en el desarrollo de jugadores automáticos (Billings, 2006).

- **Sistemas basados en el conocimiento:** Estos sistemas combinan sistemas deterministas basados en reglas, junto con sistemas de simulación de manos y oponentes. Estos sistemas generan estadísticas sobre el estado de la partida y en función de la regla que disparen tales datos toman una decisión u otra. El único jugador automático que ha conseguido superar a un humano medio en el juego es POKI, y utiliza un sistema de este tipo para jugar. Juega en mesas con limite bajo¹. Sin embargo estos sistemas presentan las carencias inherentes de los sistemas basados al conocimiento (Billings et al. 2002), (Schaeffer et al. 1999).
- **Sistemas basados en teoría de juegos:** Estos sistemas basan el juego de los jugadores automáticos en estrategias en el que este balanceado el juego en función de jugadas que pretenden engañar al oponente (faroles y trampas) con las acciones de juego (subir, igualar o tirarse). Estos sistemas utilizan una reformulación del juego para simplificar el problema, reduciendo al máximo su tamaño (Billings et al. 2003) (Gilpin y Saldholm 2006).
- **Sistemas basados en búsqueda informada:** Estos sistemas basan su juego en el conocimiento adquirido del juego de sus oponentes. Realizan un modelado del oponente, explotando aquellas características del oponente para obtener beneficio. Estos sistemas (VEXBOT) pueden llegar a alcanzar un nivel de conocimiento que les permite superar a sus oponentes por un amplio margen, suponiendo una importante amenaza a jugadores profesionales (Billings et al. 2004).

El sistema que se va a desarrollar utiliza algunos de las técnicas que se han visto en esta sección, tales como la generación de estadísticas para determinar el estado de la mesa.

¹ Partidas en las que el valor de las fichas está comprendido entre 0.25\$ y 1\$ (PokerStars).



1.2 Objetivos

El objetivo principal de este proyecto es obtener, mediante algoritmos genéticos, un jugador de póker que juegue utilizando una estrategia optimizada para ganar, es decir que rentabilice pérdidas y ganancias obteniendo beneficios.

Para conseguir el objetivo principal del proyecto se han de cumplir primero sub-objetivos.

- **Definición de la solución:** Se ha de desarrollar un sistema o una estructura que sea capaz de jugar al póker garantizando un juego rentable, es decir, que minimice pérdidas y maximice ganancias. Cada estructura representa una solución diferente, que puede estar más o menos cerca de la solución que se quiere encontrar.
- **Herramienta de evolución:** Se ha de implementar una herramienta que consiga evolucionar soluciones para el problema que se acerquen a los criterios esperados. Esta herramienta ha de ser capaz de obtener una solución aproximada a la esperada.
- **Codificación:** Se ha de encontrar una codificación que contemple todas las soluciones. A su vez ha de poder diferenciar estados dentro de una partida de póker, tomando una decisión u otra dependiendo de la situación en la que se encuentre. Adicionalmente esta codificación ha de ser suficiente ligera como para que se pueda trabajar con ella disponiendo de una memoria limitada.

Función de evaluación: Es una función que asigna un valor numérico a cada solución. En este caso la función de evaluación tendrá que maximizar ganancias minimizando pérdidas, por tanto incorpora un compromiso entre ambos objetivos. La función de evaluación orienta la búsqueda, por consiguiente, impacta notablemente en la calidad de la heurística que el algoritmo evolutivo pueda conseguir. Como requisito adicional, es conveniente que su cálculo sea lo más eficiente posible ya que una parte importante de la carga computacional del algoritmo evolutivo recae en su cálculo. Así mismo, la función de evaluación ha de ser capaz de co-evolucionar con la población de soluciones para que se mantenga cierto nivel de dificultad y fuerce la evolución.

- **Sistema de visualización:** Se ha de obtener un sistema para que, mediante un software de visualización de partidas de póker, se pueda realizar un seguimiento con todo el nivel de detalle posible para analizar las estrategias de los jugadores.



- **Juego de entrenadores:** Se ha de tener un juego de entrenadores con una forma de jugar coherente con los que entrenar a los nuevos individuos. Estos entrenadores han de mejorar su juego en función del nivel de juego del individuo que se está evolucionando. El juego de entrenadores en un principio iba a consistir en un conjunto de jugadores pre-programados, basados en reglas. Este sistema (como se explicara más adelante) tiene los problemas de ser fijo, lo que hace que en algún momento quede sobrepasado (en termino de ganancias) por los individuos y que el algoritmo se especialice contra este conjunto, perdiendo generalidad.

Una vez cumplidos todos los sub-objetivos anteriores, se puede decir que el proyecto esta ha cumplido todos los objetivos necesarios para la posible obtención de una estrategia de juego mejorada (partiendo de un jugador aleatorio).



1.3 Introducción al juego del póker

El juego del póker, es un juego de cartas en el que cada realiza apuestas en función de sus cartas y de las decisiones que han tomado el resto de jugadores.

Para jugar se necesitan al menos dos jugadores y una baraja francesa de 52 cartas (hay variantes del juego que requieren menos cartas pero este trabajo se desarrolla sobre una variante de 52). Los jugadores que lleguen al final de una mano² mostrarán sus cartas, y el que tenga la combinación más alta se lleva el bote acumulado (en el caso de empate se reparte). Si un jugador consigue echar al resto de jugadores, este jugador se lleva automáticamente el bote que se acumulado en la mesa.

El juego del póker se trata de un juego de azar, con un componente estratégico y psicológico muy importante. Los jugadores profesionales utilizan la estrategia y la capacidad de adivinar la jugada para obtener beneficios en el juego.

El póker tradicionalmente ha sido un tema tabú en la mayoría de sociedades, un juego clandestino el que se gastaba el dinero gente de dudosa reputación. Sin embargo con la aparición de la web 2.0, de las casas de póker on-line y de jugadores de póker amateur que ganan World series of Poker (Chris "MoneyMaker", jugador de póker on-line amateur gana las World series of Poker en 2003), se ha disparado el número de jugadores de póker, siendo uno de los juegos on-line más populares a nivel mundial.

Actualmente en muchas partes del mundo el póker es un deporte de competición, en el que hay grandes estrellas y mucho dinero de por medio.

Se han realizado muchos estudios y desarrollos para generar jugadores de póker automáticos. Actualmente se conocen algunas estrategias ganadoras, pero estas estrategias suelen ser muy pobres, basándose en únicamente en "apostar a lo seguro", como la técnica "Short stack", que consiste en jugar con poco dinero manos muy fuertes. Estas estrategias suelen ser reconocidas por las casas de póker on-line y los jugadores que utilizan estas tácticas son expulsados de las partidas. Hay dos aproximaciones para obtener jugadores automáticos de póker (pokerbot). La primera consiste en determinar las reglas de juego que utilizan los jugadores expertos de póker y programar al pokerbot con ellas. Aunque menos numerosa que la basada en reglas, la aplicación de técnicas de aprendizaje e inteligencia artificial es una segunda aproximación para obtener pokerbots que actualmente crece en número de propuestas (Billings, 2006).

Los enfoques de aprendizaje automático, pueden, o bien tratar de meter en el sistema estas reglas de juego, o por el otro lado, tratar de generar ese

² Mano: Una mano equivale a una ronda de juego en el póker. Este concepto se explicara en la sección "2.3 Reglas del Póker".



Universidad
Carlos III de Madrid
www.uc3m.es

conocimiento. El sistema desarrollado trata de realizar esto último con el fin de generar un jugador automático (Teófilo y Reis, 2011).



1.4 Aprendizaje automático vs robots pre programados

A la hora de abordar el problema de generar un jugador de póker automático, se pueden elegir dos caminos principales: crear un robot pre programado o generar un robot mediante técnicas de aprendizaje automático.

Si se aborda desde el primer camino, el jugador que resulte tendrá una estrategia de juego potencialmente competitiva. Sin embargo generar un jugador de esta manera tiene sus inconvenientes.

Para programar un jugador de póker competitivo se necesita un gran conocimiento sobre el póker, para poder abarcar todas las posibilidades y con un extenso conocimiento sobre estrategias de juego. Adicionalmente se ha de contemplar en el código cualquier posibilidad, y si se quiere llegar a una estrategia óptima se ha de invertir mucho tiempo y esfuerzo en prever el mayor número de situaciones posibles en la que el robot pueda llegar a estar. Finalmente se ha de tener un sistema de pruebas muy costoso en tiempo, ya que se ha de probar contra otros jugadores humanos y el rito de juego puede ser limitado. Para cuando se puedan evaluar todas las situaciones en las que se pueda encontrar el robot pueden haber pasado meses. Programar un robot que juegue al póker con una estrategia óptima, por ende, se convierte en una labor de mucho tiempo y esfuerzo, con un código con muchos cientos de reglas.

En el caso de que se aborde con técnicas de aprendizaje automático, el jugador de póker que se puede obtener, si se aplican bien las técnicas puede ser un jugador que pueda rentabilizar las ganancias frente a las pérdidas y con la capacidad para adaptarse a nuevas condiciones de juego.

Para conseguir un jugador con técnicas de aprendizaje automático no se necesita un conocimiento tan extenso del póker (cierto conocimiento es indispensable), en el caso de los algoritmos genéticos se introduce indirectamente en la representación de la solución y en la función de evaluación. Los inconvenientes que presenta este sistema son: necesidad de una herramienta que permita entrenar a los jugadores, el planteamiento de una codificación con suficiente profundidad como para poder representar fielmente un estado dado, necesidad de entrenadores contra los que se pueda entrenar a los jugadores y un enorme espacio de búsqueda que puede hacer inviable alcanzar soluciones competitivas en tiempos asumibles. Sin embargo comparativamente con la otra opción se necesita mucho menos tiempo para obtener un jugador que pueda jugar de una manera relativamente coherente, pudiendo además, utilizar conjuntos de situaciones mucho mas grandes con los que probar las capacidades de los individuos. Adicionalmente la evaluación de los jugadores no tiene el sesgo de tenerlos que entrenar con humanos (cansancio, estado anímico, motivación... etc.). Sin embargo, la validación de los jugadores generados con este sistema debería hacerse con jugadores humanos.



Este trabajo consiste en la generación de un jugador de la nada, es decir, sin ningún tipo de conocimiento sobre el póker introducido directamente, con una estrategia que rentabilice las ganancias y minimizando pérdidas. En los siguientes apartados se describirán todos los pasos que se han seguido para obtener un jugador con estas características.



2. Descripción del dominio

En esta sección se procederá a describir el dominio sobre el que se ha desarrollado el sistema. El dominio en cuestión es el juego del póker.

En las siguientes secciones se detallarán todos los aspectos del juego del póker, en qué consiste, cuáles son las reglas del juego, y qué factores son importantes a la hora de desarrollar un jugador automático competitivo.

2.1 Introducción

Para comenzar se realizará una descripción del juego del póker y de algunas de sus variantes. Se detallará cuando apareció, en qué consiste y cuál es su estado actual. Se explicarán las reglas de póker y cuáles son las combinaciones que se pueden conseguir en una mano.

Adicionalmente se explicará la variante sobre la que se ha implementado el sistema.

Se realizará una explicación de los factores clave que hay que tener en cuenta a la hora de desarrollar un jugador de póker.



2.2 Juego del póker

El juego del póker es un juego en el que participan de dos a diez jugadores en una misma mesa. Se emplea una baraja francesa de 52 y un set de fichas para representar las apuestas. La baraja francesa contiene 13 cartas numeradas del 1 al 10 y J, Q y K de cuatro palos diferentes (tréboles, picas, diamantes y corazones).

El juego del póker tiene un gran porcentaje de azar, sin embargo, el azar es el mismo para todos y lo que realmente diferencia un jugador bueno de otro malo es su estrategia de juego. El manejo del azar ha de tenerse muy en cuenta en el desarrollo de la técnica, ya que ha de estar correctamente repartido entre los jugadores, para que no aparezcan ventajas azarosas (rotando las cartas y las posiciones de los jugadores).

El juego del póker aparece en Francia, en 1829 ya era un juego muy conocido en Estados Unidos. El juego sin embargo ha ido sufriendo diferentes modificaciones a lo largo del tiempo hasta tomar la forma que conocemos actualmente.

El juego del póker consiste en ir apostando a lo largo de la mano especulando sobre las posibilidades de tener una combinación mejor que el resto de jugadores.

El trabajo que vamos a realizar consiste en generar un jugador automático, que en función de las cartas de las que dispone y de las acciones del resto de jugadores, sea capaz de tener un juego en el que pueda maximizar ganancias y minimizar pérdidas.



2.3 Reglas del póker

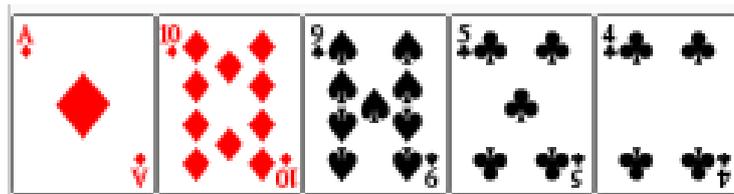
El póker es un juego que se juega por manos. Cada mano que se reparte es independiente de la otra (sin tener en cuenta la diferencia de dinero debida a las apuestas). En cada mano cada jugador recibe un número de cartas (dependiendo de la variante que se esté jugando) y en función de las cartas que tiene y de las apuestas del resto de jugadores ha de tomar una decisión. Las decisiones que toma se pueden reducir a tres: se tira, es decir, se da por vencido y decide no continuar con la mano perdiendo el dinero apostado; iguala la apuesta que hay sobre la mesa; sube la apuesta (dependiendo de la variante puede subir una a cantidad indefinida o una estipulada por la mesa en la que se está jugando).

Si un jugador se queda solo, es decir, el resto de jugadores han tirado sus cartas, este jugador gana automáticamente la mano sin necesidad de enseñar las cartas que tenía (revelar información al resto de jugadores, puede ayudarles a mejorar su juego).

En el caso de que se haya terminado la fase de apuestas, si ha quedado más de un jugador en la mesa, el último jugador que haya subido debe mostrar sus cartas. En caso de que el otro jugador tenga una combinación mejor también ha de enseñarlas, en caso contrario no tiene por qué hacerlo. El jugador con la mejor combinación gana la mano y se lleva el bote.

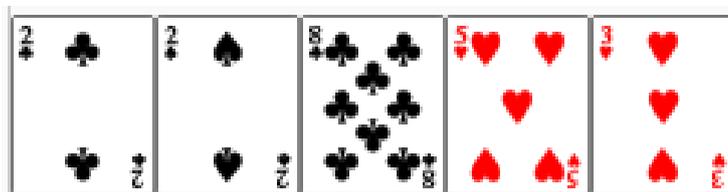
2.3.1 Combinaciones

- **Carta alta:** Combinación más baja. La combinación se representa con la carta más alta de la que dispone un jugador.



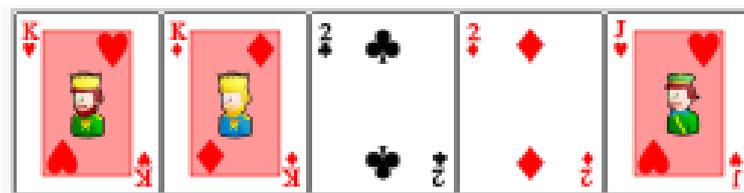
En este caso la carta más alta sería el as de picas.

- **Pareja:** La combinación se representa con dos cartas del mismo número.



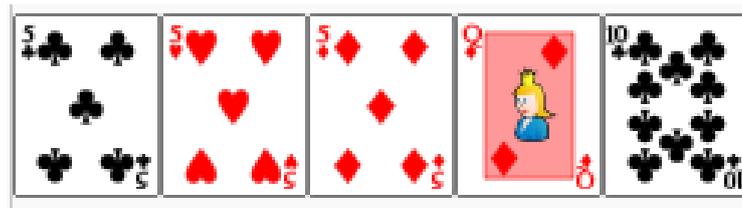
En este caso la pareja sería una pareja de doses.

- **Dobles parejas:** La combinación se representa con dos parejas de números diferentes. La pareja más representativa es la de números más altos, siendo la de ases la más alta y la de doses la más baja.



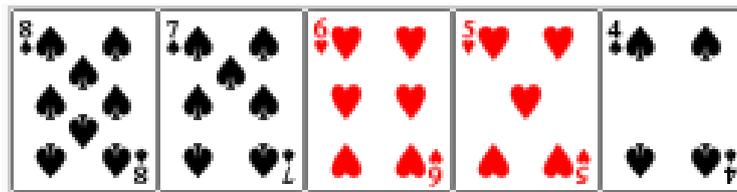
En este caso las dobles parejas serian de K's y doses

- **Trío:** La combinación se representa con tres cartas del mismo número.



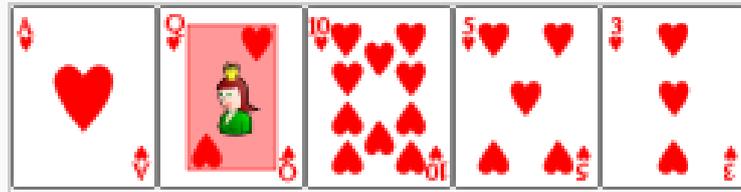
En este caso el trío sería el de cincos.

- **Escalera:** La combinación se representa con cinco cartas con números consecutivos. La carta más representativa es la carta más alta. En el caso de tener una escalera del As al cinco, la carta más representativa es el cinco, sin embargo, en caso de tener una escalera del 10 al As la carta más representativa es el As.



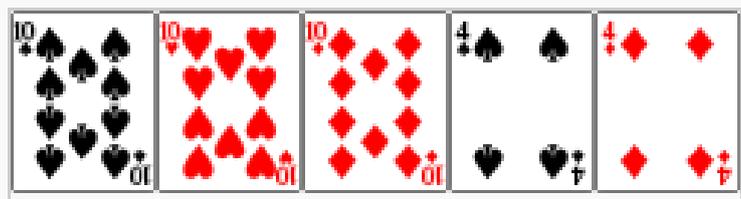
En este caso la escalera seria del cuatro a ocho.

- **Color:** Esta combinación se representa con cinco cartas del mismo palo. La carta más representativa es la carta con el número más alto. En el caso de haber empate con la carta más representativa, se compara la siguiente carta con el número más alto. Esta comparación se repite hasta que se han comparado todas las cartas que forman el color.



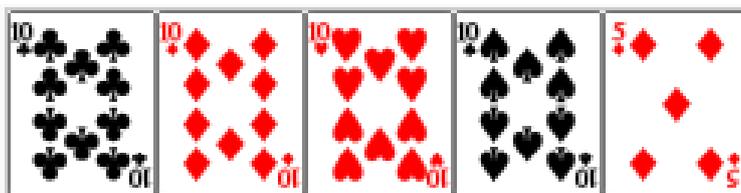
En este caso el color sería color de corazones al As.

- **Full:** Esta combinación se representa con un trío y una pareja. La parte más representativa es el trío y luego la pareja.



En este caso el full es de dieces cuatros.

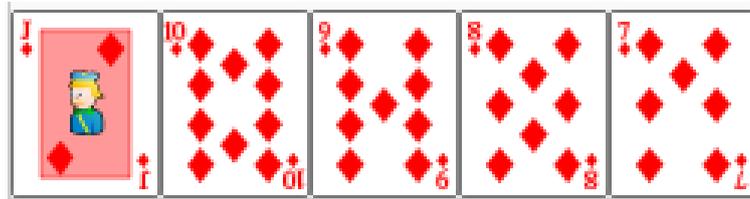
- **Póker:** Esta combinación se representa con cuatro cartas del mismo número.



En este caso el póker es de dieces.

- **Escalera de color:** Esta combinación se representa con una escalera en la que todas las cartas que la componen son de un palo. La carta más representativa es la carta más alta. En el caso de tener una escalera del As al cinco, la carta más representativa

es el cinco, sin embargo, en caso de tener una escalera del 10 al As la carta más representativa es el As.



En este caso la escalera de color es del siete a la J.

Numero de combinaciones en una baraja.

Carta alta	1.302.540
Pareja	1,098,240
Dobles parejas	123,552
Trío	54,912
Escalera	10,240
Color	5,148
Full	3,744
Póker	624
Escalera de color	40

El número total de combinaciones es 2.598.960

En caso de haber un empate en las combinaciones, se comparan las siguientes cartas en orden descendente según su número que están fuera de la combinación hasta que la suma de estas nuevas cartas y las que están dentro de la combinación sumen cinco. Por ejemplo, si se tiene carta alta se compararían las siguientes cuatro cartas con números más altos. En el caso de que la combinación sea de cinco cartas, no se compararían más cartas.

En el caso de haber empate los botes se reparten.



2.4 Variante Texas Hold'em

El juego del póker tiene muchas variantes diferentes, cada una tiene un juego de reglas distinto, se utilizan más o menos cartas, algunas se enseñan, otras no (Sklansky 1994).

El sistema que se ha modelado está pensado para jugar la variante de póker Texas Hold'em (Sklansky 1997). Esta variante se juega con hasta cinco cartas descubiertas (depende de la fase, que se explicara en posteriores apartados) y dos cubiertas. Las cinco cubiertas son cartas que puede utilizar cualquier jugador para realizar la mejor combinación posible. Cada jugador dispone de hasta siete cartas para realizar la mejor combinación posible utilizando cinco de ellas.

La variante del Texas hold'em se juega en cuatro fases diferentes. En cada fase cada jugador apuesta en función de las perspectivas de ganar que tiene (o de cómo quiere engañar a sus rivales) comenzando por el jugador que puso la ciega pequeña (en el preflop es el siguiente a la ciega grande).

Cada fase termina cuando o bien solo queda un jugador, lo que conllevaría a terminar la mano, o cuando el numero de apuestas de todos los jugadores es el mismo.

Cada mano termina cuando se llega a la fase final o si algún jugador ha conseguido quedarse solo en la mesa de juego.

Cuando finaliza la mano si ha quedado más de un jugador con cartas en la mesa, estos muestran su combinación. El jugador con la combinación más alta gana. En caso de solo quedar un jugador en juego, este jugador gana automáticamente el bote que haya sobre la mesa.

Esta variante de juego es de las más populares actualmente, con la que millones de jugadores de póker apuestan diariamente tanto en mesas de juego online, torneos de póker en casinos y salas de juego como en torneos patrocinados televisados. Así mismo tiene especial interés (especialmente la variante con limite fijo), ya que se pueden desarrollar estrategias ganadoras, mientras que con otras variantes puede ser mucho más complicado.

Las estrategias para cada tipo de variante del póker puede variar enormemente, como demuestran los textos específicos para cada tipo de juego: (Phil Gordon 2005), torneos (Harrington y Robertie 2004), Juego con límite (Sklansky y Malmuth 1999) o los que tratan de establecer compendios de similitudes y diferencias entre los distintos estilos (Brunson et al. 2002), (Sklansky 1994)



2.5 Variante con límite fijo

La variante con límite fijo consiste en limitar las apuestas a una cantidad fija cada vez que se apuesta (Sklansky y Malmuth 1999). Esta limitación viene dada por la mesa, luego cuando un jugador decide realizar una apuesta solo puede tomar la decisión de subir igualar o tirarse, pero no puede decidir cuánto quiere subir.

Durante las primeras dos fases (preflop y flop) la cantidad máxima con la que puedes subir una apuesta es igual a la ciega grande. Durante las dos últimas fases (turn y river) la cantidad máxima que se puede subir cada vez es igual al doble de la ciega grande.

El número máximo de veces que se puede subir una apuesta durante cada fase es de cuatro veces. En el caso de que todos los jugadores hayan subido cuatro veces sus apuestas se termina esa fase automáticamente y se pasa a la siguiente (en el caso del river, con la variante en la que se desarrolla el proyecto, se termina la mano).

La variante con límite fijo, es una variante muy interesante desde el punto de vista de la inteligencia artificial, ya que se pueden plantear los estados del autómatas que se está evolucionando en función del número de subidas, ya que el número de subidas puede representar un estado diferente.



2.6 Posiciones

En el juego del póker cada jugador ocupa una posición alrededor de una mesa formando un círculo. Cada mano que se reparte la ficha del Dealer se mueve una posición en sentido de las agujas del reloj. La ficha del Dealer marca al jugador que ocupa la posición de Dealer. Tradicionalmente el Dealer es el jugador que se encarga de repartir esa mano, aunque en los casinos y en las salas de póker online, es un sistema o una persona que no interviene en el juego, la que se encarga de repartirlas.

Las posiciones en el juego del póker están dispuestas en función de la posición del Dealer. El último jugador que toma su decisión es el Dealer y el primero es el de su derecha (mirando la mesa desde arriba) y el turno de cada jugador llega en sentido de las agujas del reloj.

La posición en el juego del póker es un factor fundamental a la hora de tomar una decisión u otra. La posición más privilegiada es la del Dealer, puesto que puede tomar su decisión en función de las decisiones del resto, con una perspectiva mucho más amplia.

En todas las fases salvo en la primera (preflop) el turno de cada jugador sigue el orden descrito previamente. En el preflop sin embargo, el primero en hablar es el siguiente al jugador que ocupa la posición de ciega grande, dos posiciones a la derecha del Dealer, y la ciega grande es la última en jugar.

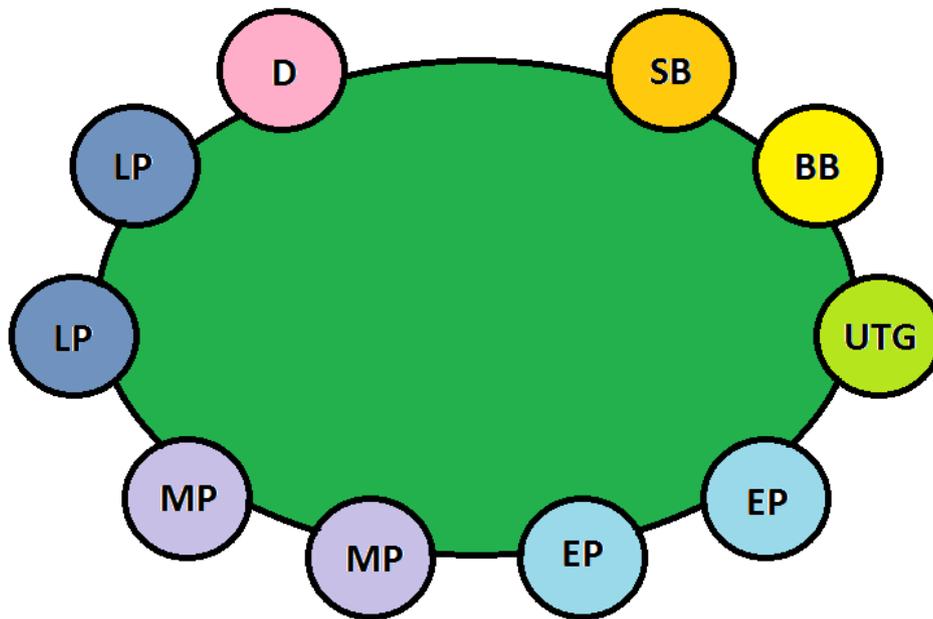
A continuación se describen cada una de las posiciones de una mesa de diez jugadores.

- **Ciega pequeña:** Esta posición es la penúltima en hablar en la fase del preflop y la primera en el resto de fases. Esta posición se ve obligada a poner una cantidad de dinero igual a la mitad de la ciega grande antes de recibir cualquier carta. Esta posición puede jugar un rango de cartas bastante amplio ya que ha puesto dinero en la mesa y únicamente tiene que poner media ciega para continuar en la mano.
- **Ciega grande:** Esta es la última fase del preflop y la segunda en hablar en el resto de fases. Esta posición se ve obligada a poner una cantidad igual a la ciega grande (marcada por la mesa) antes de escribir cualquier carta. Esta es la posición que puede jugar un rango de cartas más amplio después del Dealer ya que ha puesto dinero en la mesa y no tiene que añadir más dinero para muchas de las manos.



- **Primero en hablar o "Under the gun":** Esta es la posición que habla primero en la fase del preflop. Esta posición es la que juega un rango de cartas más limitado, puesto que como tiene que hablar primero ha de tener buenas cartas si quiere participar en la mano.
- **Posición temprana o "Early position":** Esta posición al igual que el UTG solo puede jugar un rango de cartas limitado aunque no tanto como su predecesor.
- **Posición media o "Middle position":** Esta posición juega un rango de cartas medio, pudiendo tomar decisiones más arriesgadas que las de posiciones tempranas puesto que tienen más perspectiva.
- **Posición tardía o "Late position":** Estas posiciones pueden jugar un rango de cartas más amplio todavía ya que tienen a muchos jugadores antes y pueden prever la jugada de los anteriores.
- **Dealer:** Esta posición es la que habla la última. Esta posición es la que juega el rango de cartas más amplio pudiéndose arriesgar mucho más que cualquier otra posición, ya que puede ver todas las decisiones que han tomado el resto de jugadores.

A continuación se presenta un diagrama en el que se muestran las iniciales en inglés de cada posición al rededor de una mesa de póker.



2.7 Fases

En esta sección se describirán las cuatro fases que hay en cada mano de póker en su variante de Texas hold'em.

La fase en la que te encuentras es un factor imprescindible en el juego del póker, puesto que determina con cuántas cartas puede hacer una combinación, cuántas cartas quedan por repartir y cuáles son tus probabilidades de éxito en la mano que estás jugando.

Dependiendo de en qué fase te encuentras puedes con la misma combinación desde subir una apuesta hasta tirarte directamente sabiendo que con esta combinación no vas a ganar la mano.

La fase es un factor clave en el modelado del autómatas que se evoluciona para conseguir un jugador de póker efectivo.

2.7.1 Preflop

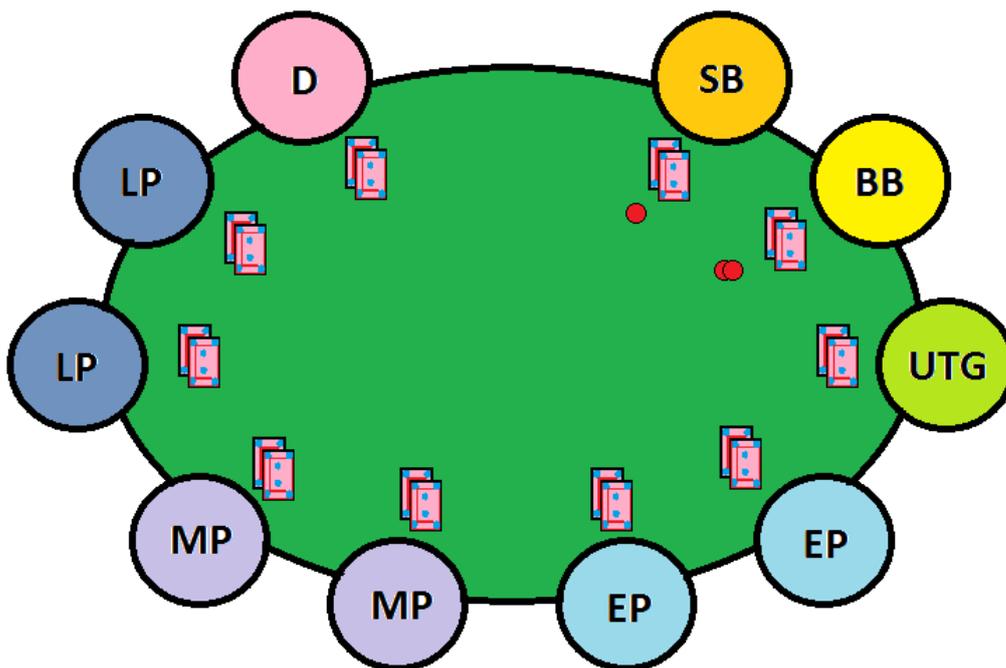
La fase de preflop es la primera fase de cada mano del juego del póker.

Al principio de esta fase la posición de la ciega pequeña antes de recibir sus cartas ha de poner una cantidad de fichas igual a la mitad de la ciega grande. La ciega grande ha de hacer lo mismo pero con una cantidad de fichas igual a la ciega grande.

A continuación a todos los jugadores se les reparte dos cartas boca abajo que son propias de cada jugador y el resto no debe conocer.

La combinación más alta que se puede realizar en la fase del preflop es una pareja de ases.

A continuación se muestra un diagrama con la disposición de los jugadores.



En esta fase el primero que habla es la posición a la derecha de la ciega grande, la posición del UTG.

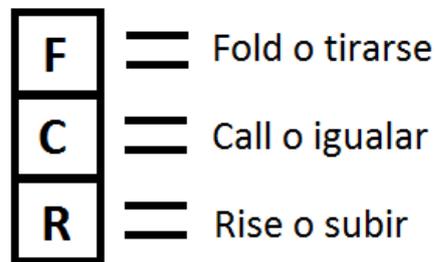
Cada jugador toma su decisión en función de las cartas que tiene y de las decisiones que van tomando el número de jugadores. La fase del preflop finaliza cuando todos los jugadores han igualado apuestas, cuando sólo queda un jugador con cartas o bien todos los jugadores han alcanzado cuatro subidas.



En esta variante del póker con límite fijo es normal utilizar tablas de preflop. Estas tablas dictan qué decisión tomar en cada posición en función de las cartas de las que dispones y del número de subidas que han realizado en resto de jugadores.

A continuación se muestran las tablas de preflop de cada una de las posiciones.

Las decisiones se dividen en tres tipos: Tirarse representado por una F de fold, igualar que está representado por una C de call y subir que está representado por una R de rise.



A continuación se muestra un diagrama mostrando el diagrama de colores que representan la decisión que se va a realizar dependiendo de las cartas, de la posición y del número de subidas. El índice de cuadrado va del uno al cuatro e indica el número de subidas que se han realizado.

Las tablas preflop, las utilizo por primera vez David Sklansky (jugador de póker profesional y escritor de varios libros de póker), lo que hizo crecer su popularidad.

Los colores están ordenados en función de la potencia de las cartas, es decir, las cartas menos potentes tienen relacionado una jugada con el color blanco, las más potentes tienen asociado el color rosa.



F	F	F	F
C	F	F	F
C	C	F	F
R	F	F	F
R	C	F	F
R	R	C	C
R	R	R	C

A continuación se muestran las tablas del preflop para cada posición y combinación de cartas. Las letras en rojo indican que son del mismo palo, las que están en negro indican diferente palo (Gonzalo García-Pelayo, Diego Pradera, Óscar García-Pelayo, 2009).



Tabla para la ciega pequeña.

AA	AK	AQ	AJ	AT	A9	A8	A7	A6	A5	A4	A3	A2
AK	KK	KQ	KJ	KT	K9	K8	K7	K6	K5	K4	K3	K2
AQ	KQ	QQ	QJ	QT	Q9	Q8	Q7	Q6	Q5	Q4	Q3	Q2
AJ	KJ	QJ	JJ	JT	J9	J8	J7	J6	J5	J4	J3	J2
AT	KT	QT	JT	TT	T9	T8	T7	T6	T5	T4	T3	T2
A9	K9	Q9	J9	T9	99	98	97	96	95	94	93	92
A8	K8	Q8	J8	T8	98	88	87	86	87	84	83	82
A7	K7	Q7	J7	T7	97	87	77	76	75	74	73	72
A6	K6	Q6	J6	T6	96	86	76	66	65	64	63	62
A5	K5	Q5	J5	T5	95	85	75	65	55	54	53	52
A4	K4	Q4	J4	T4	94	84	74	64	54	44	43	42
A3	K3	Q3	J3	T3	93	83	73	63	53	43	33	32
A2	K2	Q2	J2	T2	92	82	72	62	52	42	32	22



Tabla para la ciega grande.

AA	AK	AQ	AJ	AT	A9	A8	A7	A6	A5	A4	A3	A2
AK	KK	KQ	KJ	KT	K9	K8	K7	K6	K5	K4	K3	K2
AQ	KQ	QQ	QJ	QT	Q9	Q8	Q7	Q6	Q5	Q4	Q3	Q2
AJ	KJ	QJ	JJ	JT	J9	J8	J7	J6	J5	J4	J3	J2
AT	KT	QT	JT	TT	T9	T8	T7	T6	T5	T4	T3	T2
A9	K9	Q9	J9	T9	99	98	97	96	95	94	93	92
A8	K8	Q8	J8	T8	98	88	87	86	87	84	83	82
A7	K7	Q7	J7	T7	97	87	77	76	75	74	73	72
A6	K6	Q6	J6	T6	96	86	76	66	65	64	63	62
A5	K5	Q5	J5	T5	95	85	75	65	55	54	53	52
A4	K4	Q4	J4	T4	94	84	74	64	54	44	43	42
A3	K3	Q3	J3	T3	93	83	73	63	53	43	33	32
A2	K2	Q2	J2	T2	92	82	72	62	52	42	32	22



Tabla para la posición temprana.

AA	AK	AQ	AJ	AT	A9	A8	A7	A6	A5	A4	A3	A2
AK	KK	KQ	KJ	KT	K9	K8	K7	K6	K5	K4	K3	K2
AQ	KQ	QQ	QJ	QT	Q9	Q8	Q7	Q6	Q5	Q4	Q3	Q2
AJ	KJ	QJ	JJ	JT	J9	J8	J7	J6	J5	J4	J3	J2
AT	KT	QT	JT	TT	T9	T8	T7	T6	T5	T4	T3	T2
A9	K9	Q9	J9	T9	99	98	97	96	95	94	93	92
A8	K8	Q8	J8	T8	98	88	87	86	87	84	83	82
A7	K7	Q7	J7	T7	97	87	77	76	75	74	73	72
A6	K6	Q6	J6	T6	96	86	76	66	65	64	63	62
A5	K5	Q5	J5	T5	95	85	75	65	55	54	53	52
A4	K4	Q4	J4	T4	94	84	74	64	54	44	43	42
A3	K3	Q3	J3	T3	93	83	73	63	53	43	33	32
A2	K2	Q2	J2	T2	92	82	72	62	52	42	32	22



Tabla para la sexta posición.

AA	AK	AQ	AJ	AT	A9	A8	A7	A6	A5	A4	A3	A2
AK	KK	KQ	KJ	KT	K9	K8	K7	K6	K5	K4	K3	K2
AQ	KQ	QQ	QJ	QT	Q9	Q8	Q7	Q6	Q5	Q4	Q3	Q2
AJ	KJ	QJ	JJ	JT	J9	J8	J7	J6	J5	J4	J3	J2
AT	KT	QT	JT	TT	T9	T8	T7	T6	T5	T4	T3	T2
A9	K9	Q9	J9	T9	99	98	97	96	95	94	93	92
A8	K8	Q8	J8	T8	98	88	87	86	87	84	83	82
A7	K7	Q7	J7	T7	97	87	77	76	75	74	73	72
A6	K6	Q6	J6	T6	96	86	76	66	65	64	63	62
A5	K5	Q5	J5	T5	95	85	75	65	55	54	53	52
A4	K4	Q4	J4	T4	94	84	74	64	54	44	43	42
A3	K3	Q3	J3	T3	93	83	73	63	53	43	33	32
A2	K2	Q2	J2	T2	92	82	72	62	52	42	32	22



Tabla para la séptima posición.

AA	AK	AQ	AJ	AT	A9	A8	A7	A6	A5	A4	A3	A2
AK	KK	KQ	KJ	KT	K9	K8	K7	K6	K5	K4	K3	K2
AQ	KQ	QQ	QJ	QT	Q9	Q8	Q7	Q6	Q5	Q4	Q3	Q2
AJ	KJ	QJ	JJ	JT	J9	J8	J7	J6	J5	J4	J3	J2
AT	KT	QT	JT	TT	T9	T8	T7	T6	T5	T4	T3	T2
A9	K9	Q9	J9	T9	99	98	97	96	95	94	93	92
A8	K8	Q8	J8	T8	98	88	87	86	87	84	83	82
A7	K7	Q7	J7	T7	97	87	77	76	75	74	73	72
A6	K6	Q6	J6	T6	96	86	76	66	65	64	63	62
A5	K5	Q5	J5	T5	95	85	75	65	55	54	53	52
A4	K4	Q4	J4	T4	94	84	74	64	54	44	43	42
A3	K3	Q3	J3	T3	93	83	73	63	53	43	33	32
A2	K2	Q2	J2	T2	92	82	72	62	52	42	32	22



Tabla para la octava posición.

AA	AK	AQ	AJ	AT	A9	A8	A7	A6	A5	A4	A3	A2
AK	KK	KQ	KJ	KT	K9	K8	K7	K6	K5	K4	K3	K2
AQ	KQ	QQ	QJ	QT	Q9	Q8	Q7	Q6	Q5	Q4	Q3	Q2
AJ	KJ	QJ	JJ	JT	J9	J8	J7	J6	J5	J4	J3	J2
AT	KT	QT	JT	TT	T9	T8	T7	T6	T5	T4	T3	T2
A9	K9	Q9	J9	T9	99	98	97	96	95	94	93	92
A8	K8	Q8	J8	T8	98	88	87	86	87	84	83	82
A7	K7	Q7	J7	T7	97	87	77	76	75	74	73	72
A6	K6	Q6	J6	T6	96	86	76	66	65	64	63	62
A5	K5	Q5	J5	T5	95	85	75	65	55	54	53	52
A4	K4	Q4	J4	T4	94	84	74	64	54	44	43	42
A3	K3	Q3	J3	T3	93	83	73	63	53	43	33	32
A2	K2	Q2	J2	T2	92	82	72	62	52	42	32	22



Universidad
Carlos III de Madrid
www.uc3m.es

Tabla para la novena posición.



AA	AK	AQ	AJ	AT	A9	A8	A7	A6	A5	A4	A3	A2
AK	KK	KQ	KJ	KT	K9	K8	K7	K6	K5	K4	K3	K2
AQ	KQ	QQ	QJ	QT	Q9	Q8	Q7	Q6	Q5	Q4	Q3	Q2
AJ	KJ	QJ	JJ	JT	J9	J8	J7	J6	J5	J4	J3	J2
AT	KT	QT	JT	TT	T9	T8	T7	T6	T5	T4	T3	T2
A9	K9	Q9	J9	T9	99	98	97	96	95	94	93	92
A8	K8	Q8	J8	T8	98	88	87	86	87	84	83	82
A7	K7	Q7	J7	T7	97	87	77	76	75	74	73	72
A6	K6	Q6	J6	T6	96	86	76	66	65	64	63	62
A5	K5	Q5	J5	T5	95	85	75	65	55	54	53	52
A4	K4	Q4	J4	T4	94	84	74	64	54	44	43	42
A3	K3	Q3	J3	T3	93	83	73	63	53	43	33	32
A2	K2	Q2	J2	T2	92	82	72	62	52	42	32	22



Tabla para la posición de Dealer.

AA	AK	AQ	AJ	AT	A9	A8	A7	A6	A5	A4	A3	A2
AK	KK	KQ	KJ	KT	K9	K8	K7	K6	K5	K4	K3	K2
AQ	KQ	QQ	QJ	QT	Q9	Q8	Q7	Q6	Q5	Q4	Q3	Q2
AJ	KJ	QJ	JJ	JT	J9	J8	J7	J6	J5	J4	J3	J2
AT	KT	QT	JT	TT	T9	T8	T7	T6	T5	T4	T3	T2
A9	K9	Q9	J9	T9	99	98	97	96	95	94	93	92
A8	K8	Q8	J8	T8	98	88	87	86	87	84	83	82
A7	K7	Q7	J7	T7	97	87	77	76	75	74	73	72
A6	K6	Q6	J6	T6	96	86	76	66	65	64	63	62
A5	K5	Q5	J5	T5	95	85	75	65	55	54	53	52
A4	K4	Q4	J4	T4	94	84	74	64	54	44	43	42
A3	K3	Q3	J3	T3	93	83	73	63	53	43	33	32
A2	K2	Q2	J2	T2	92	82	72	62	52	42	32	22



2.7.2 Flop

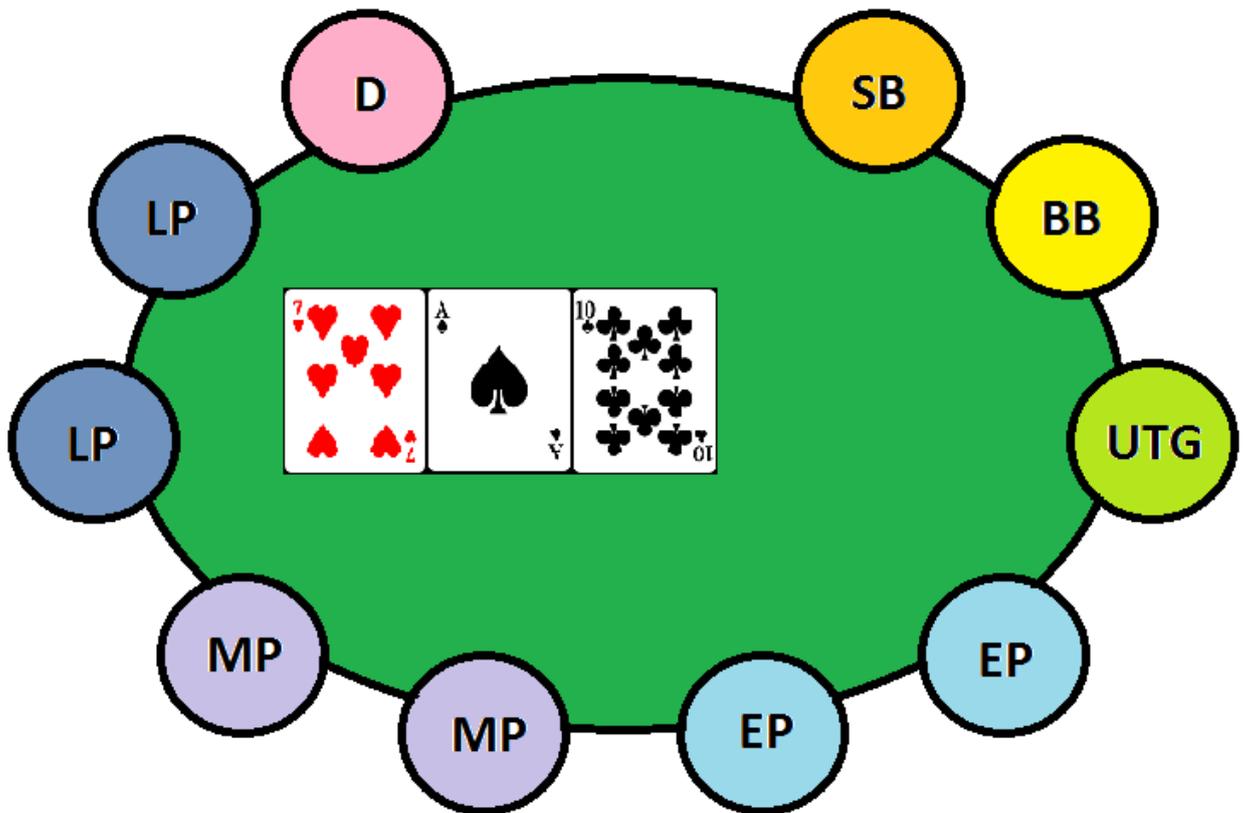
La fase del flop es la segunda fase de una mano de póker Texas hold'em. Esta fase comienza si ha quedado más de un jugador con cartas después de la fase preflop.

En esta fase se reparten tres cartas sobre la mesa. Estas cartas se reparten boca arriba y pueden ser usadas por cualquier jugador que todavía conserve las cartas de su mano. Durante esta fase se pueden realizar todas las combinaciones del póker.

Durante esta fase comienza hablando la posición de la ciega pequeña, y el último es la posición del Dealer. Cada subida en esta fase es igual a una ciega grande.

Esta fase termina cuando sólo queda un jugador en la mesa o cuando todos los jugadores han igualado sus apuestas o bien si todos los jugadores con cartas en sus manos han subido cuatro veces.

A continuación se muestra un diagrama de la mesa en la fase de flop.



2.7.3 Turn

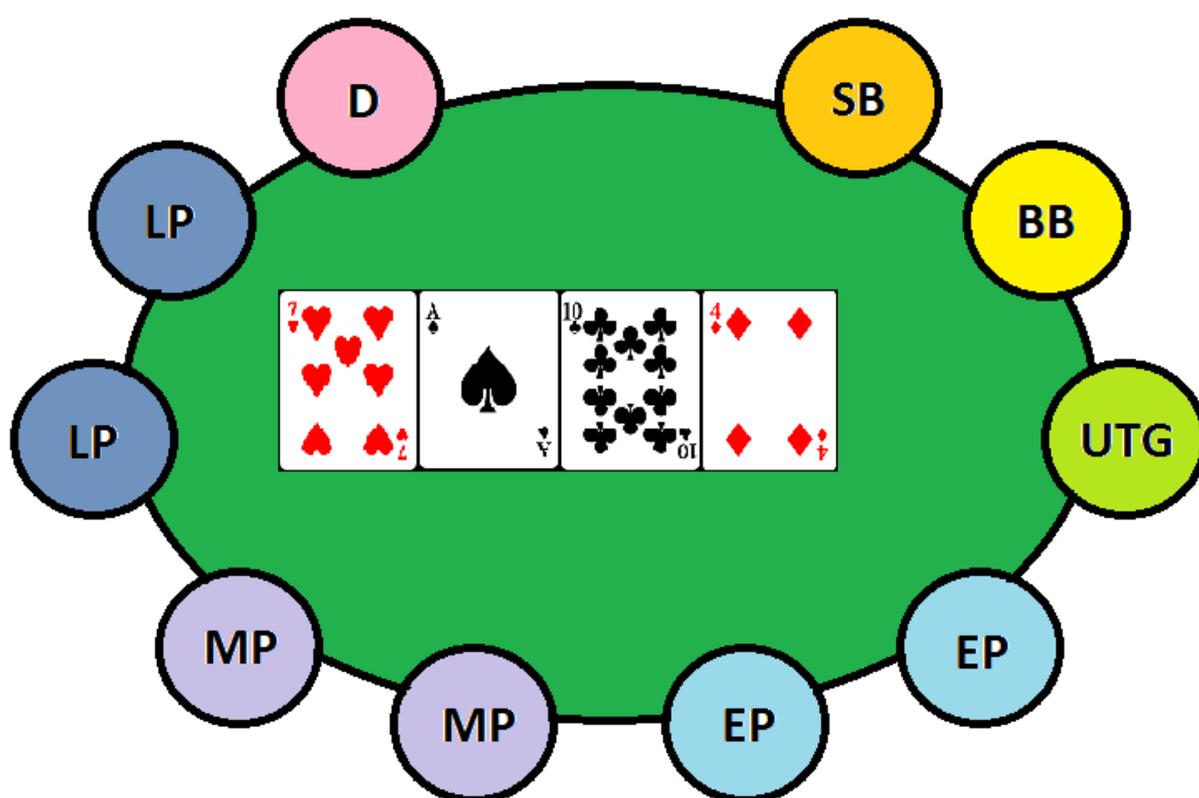
La fase del turn es la tercera fase una mano de póker. Esta fase comienza cuando se ha terminado la fase del flop con más de un jugador con cartas en su mano.

En esta fase se reparte una carta adicional sobre la mesa. Esta carta se reparte boca arriba y puede ser usada junto con las cartas del flop por cualquier jugador que todavía conserve las cartas de su mano. Durante esta fase se pueden realizar todas las combinaciones del póker.

Durante esta fase comienza hablando la posición de la ciega pequeña, y el último es la posición del Dealer. Cada subida en esta fase es igual a al doble de una ciega grande.

Esta fase termina cuando solo queda un jugador en la mesa o cuando todos los jugadores han igualado sus apuestas o bien si todos los jugadores con cartas en sus manos han subido cuatro veces.

A continuación se muestra un diagrama de la mesa en la fase de turn.



2.7.4 River

La fase del river es la tercera y última fase una mano de póker. Esta fase comienza cuando se ha terminado la fase del turn con más de un jugador con cartas en su mano.

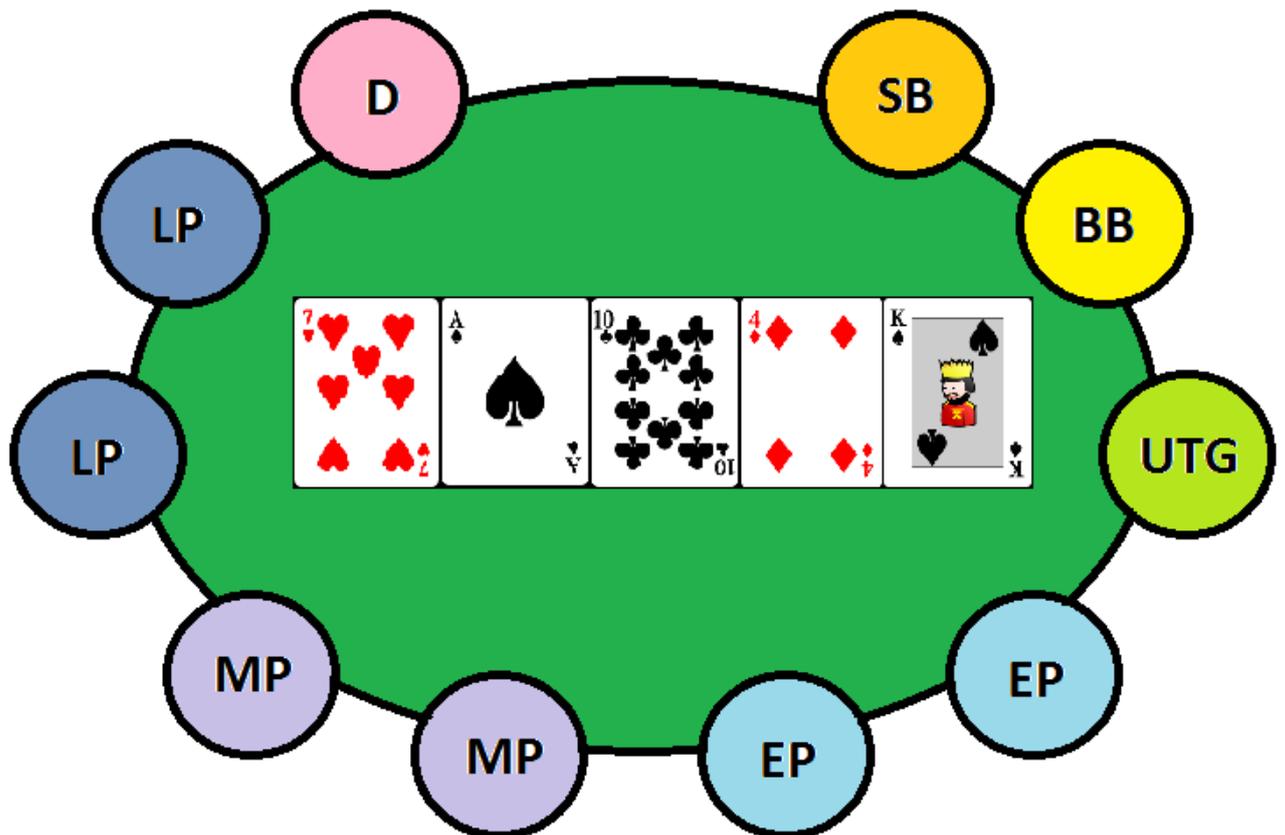
En esta fase se reparte una carta adicional sobre la mesa. Esta carta se reparte boca arriba y puede ser usada junto con las cartas del flop y del turn por cualquier jugador que todavía conserve las cartas de su mano. Durante esta fase se pueden realizar todas las combinaciones del póker.

Durante esta fase comienza hablando la posición de la ciega pequeña, y el último es la posición del Dealer. Cada subida en esta fase es igual a al doble de una ciega grande.

Esta fase termina cuando solo queda un jugador en la mesa o cuando todos los jugadores han igualado sus apuestas o bien si todos los jugadores con cartas en sus manos han subido cuatro veces.

Cuando esta fase termina se finaliza la mano y el jugador ganador se lleva el bote acumulado.

A continuación se muestra un diagrama de la mesa en la fase de river.



2.8 Apuestas

En este apartado se describe el sistema de apuestas del juego del póker y algunos conceptos relacionados con la parte económica del juego del póker.



El juego del póker es un juego de apuestas. En cada mano, cada jugador realiza una apuesta en función de la situación de la mano en la que se encuentra.

Cada jugador puede realizar una de las siguientes jugadas: subir, igualar o tirarse. La cantidad máxima de fichas que pueda subir en cada fase es de cuatro fichas. En la fase de preflop y flop cada ficha tiene un valor de una unidad de dinero y en las fases de turn y river tiene un valor de dos unidades de dinero. Las unidades de dinero las marca la mesa y son las mismas para todos los jugadores. Por ejemplo, una mesa de 2\$ - 4\$ indica que cada ficha en el preflop y en el flop cuesta 2 dólares y en el turn y en el river cuesta 4 dólares.

Un factor importante que ha de ser tenido en cuenta es la rentabilidad del bote de la jugada. La rentabilidad del bote de la jugada es el valor de fichas que has de poner en una jugada entre el valor total del bote que hay en la mesa. A mayor cantidad a depositar menos rentable es la jugada (en términos de cantidad mínima que hay que apostar entre el bote que hay en la mesa), a botes mas grandes más rentable la jugada.

La rentabilidad del bote es un factor determinante a la hora de tomar una decisión, así como el número de fichas que se han depositado tanto en la jugada actual como en las anteriores.

Adicionalmente se ha de explicar el concepto de "rake". El "rake" representa el dinero que la mesa se lleva en calidad de impuestos por jugar en la mesa.

Estos impuestos suponen un pequeño porcentaje de cada bote. Sin embargo, este impuesto ayuda a que los jugadores sean más competitivos, ya que obliga a desarrollar estrategias en las que el dinero que gana un jugador supla las perdidas por impuestos.

En el caso de que todos los jugadores jugasen igual y no hubiera impuestos, el beneficio obtenido seria 0, el conjunto de jugadores entraría en un equilibrio y no habría necesidad de seguir mejorando. Gracias a los impuestos, si se diera el caso de que todos los jugadores fueran iguales, los impuestos provocarían un resultado negativo en las ganancias de todos los jugadores (Borer et al., 2009).



2.9 Combinatoria

El juego del póker tiene un número finito de combinaciones de cartas y apuestas, así como de combinaciones que marcan la puntuación de una mano.

En la siguiente tabla mostramos el número de combinaciones para marcar la puntuación que se pueden generar con una baraja. La primera tabla muestra el número total de combinaciones, la segunda muestra el número en función de la combinación menos probable.

Carta alta	1302540	32564
Pareja	1098240	27456
Dobles parejas	123552	3088,8
Trio	54912	1372,8
Escalera	1024	25,6
Color	5148	128,7
Full	3744	93,6
Póker	624	15,6
Escalera de color	40	1

En la tabla que mostramos a continuación se muestra el número de combinaciones posibles de cartas y apuestas que pueden existir en el juego del póker. La columna de la izquierda indica el número de combinaciones de cartas, apuestas y de cartas con apuestas dependiendo de la ronda. La columna central indica el número de combinaciones en una fase de la mano dada. La columna de la izquierda indica la fase de la mano.

Combinaciones	Numero	Ronda
	2652	preflop
78923520	29760	flop
2288782080	29	turn
64085898240	28	river
Apuestas		
	4	preflop
20	5	flop
100	5	turn
500	5	river
Apuestas x Cartas		
	10608	preflop
1578470400	148800	flop
2,28878E+11	145	turn
3,20429E+13	140	river



2.10 Póker y aprendizaje automático

Este documento trata sobre el desarrollo de un jugador de póker automático utilizando técnicas de aprendizaje automático.

La técnica de aprendizaje automático que se va a utilizar para resolver este problema son los algoritmos genéticos. Mediante esta técnica y utilizando los conceptos sobre el póker explicados en los apartados anteriores se modelará el algoritmo y el individuo para poder abordar este problema y llegar a una solución aceptable.

El problema se abordará modelando las reglas de juego dentro del sistema, aplicando el sistema de fases e identificando los factores del juego para modelar la solución de forma que sea suficientemente genérica, pero sin ser tan grande que resulte inabarcable. El número de factores representados en la solución afecta directamente en el espacio de búsqueda.

El problema del póker se transformará en un problema de optimización (optimización de ganancias) para poder aplicar técnicas evolutivas para su resolución.

Si bien es cierto que para un modelado del problema sería muy interesante realizar un análisis de oponentes, éste problema se sale del proyecto que se redacta en este documento y daría suficiente trabajo como para realizar otro proyecto de una envergadura similar.

Por tanto este proyecto abordará el aprendizaje de jugadores automáticos sin tener en cuenta el oponente.

En los siguientes apartados se describirá de forma detallada el funcionamiento del algoritmo así como el modelado que se ha realizado del sistema.



3. Algoritmos evolutivos

En este apartado se describirá el funcionamiento de un algoritmo genético canónico, explicando cómo funciona y las bases teóricas en las que se fundamenta.

Así mismo se realizara una descripción de las modificaciones que se han realizado al algoritmo canónico para poderlo aplicar al problema del póker.

3.1 Introducción

Los algoritmos genéticos son técnicas de optimización que emplean una metáfora del proceso de selección natural de las especies darwinista (Charles Darwin 1859).

Estos algoritmos desarrollados sobre la década de los 60 del siglo XX (Barricelli, Nils Aall 1963) se basan en la búsqueda dirigida hacia una solución sub-óptima.

La búsqueda dirigida es un sistema mucho mas óptimo que la búsqueda no dirigida o aleatoria, puesto que las soluciones que se proponen sientan su base sobre una solución conocida.

Mediante el uso de un operador de elitismo (se describirá posteriormente) se puede demostrar que el algoritmo converge hacia una solución sub-óptima.

En los siguientes apartados se describirá el algoritmo en detalle explicando cada una de sus partes.



3.2 Algoritmos Genéticos

El algoritmo genético consiste en un algoritmo que mediante una serie de operaciones sobre una población inicial, consigue encontrar una solución sub-óptima de un problema dado.

La población de un algoritmo genético consiste en el conjunto de individuos del que dispone para probar diferentes soluciones.

Cada individuo de la población, codifica de alguna manera una solución del problema. La representación de la solución depende de la codificación de los individuos.

El algoritmo evalúa la calidad cada solución representada en la población mediante una función de evaluación. Cada ronda de evaluación de la población, junto a la aplicación de operadores (se explicaran posteriormente en qué consisten los operadores), es a lo que llamamos generación.

Cada generación, la población mejora la calidad de la solución o se mantiene igual, pero nunca empeora.

En este proyecto se ha utilizado el Algoritmo Genético Canónico como base para realizar el entrenamiento de los jugadores. En siguientes apartados se describirán las modificaciones que se han realizado para adecuarlo al problema que se trata.

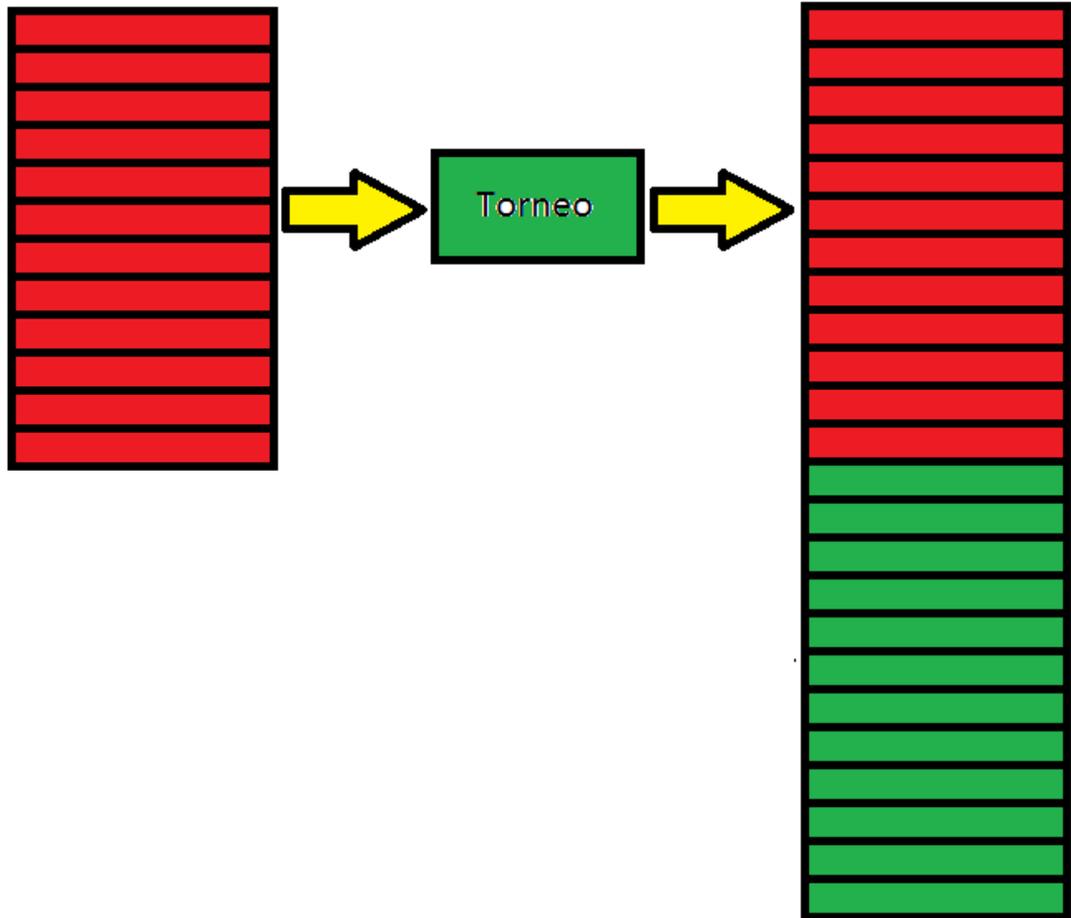
La población de la que parte el algoritmo genético, es una población que se inicializa de forma aleatoria. Cada individuo de la población representa un punto aleatorio en el espacio de búsqueda de las soluciones al problema.

Mediante una serie de operaciones la población se va acercando hacia una solución mejor en cada generación del algoritmo.

Los operadores que utiliza el algoritmo sobre la población son los siguientes:

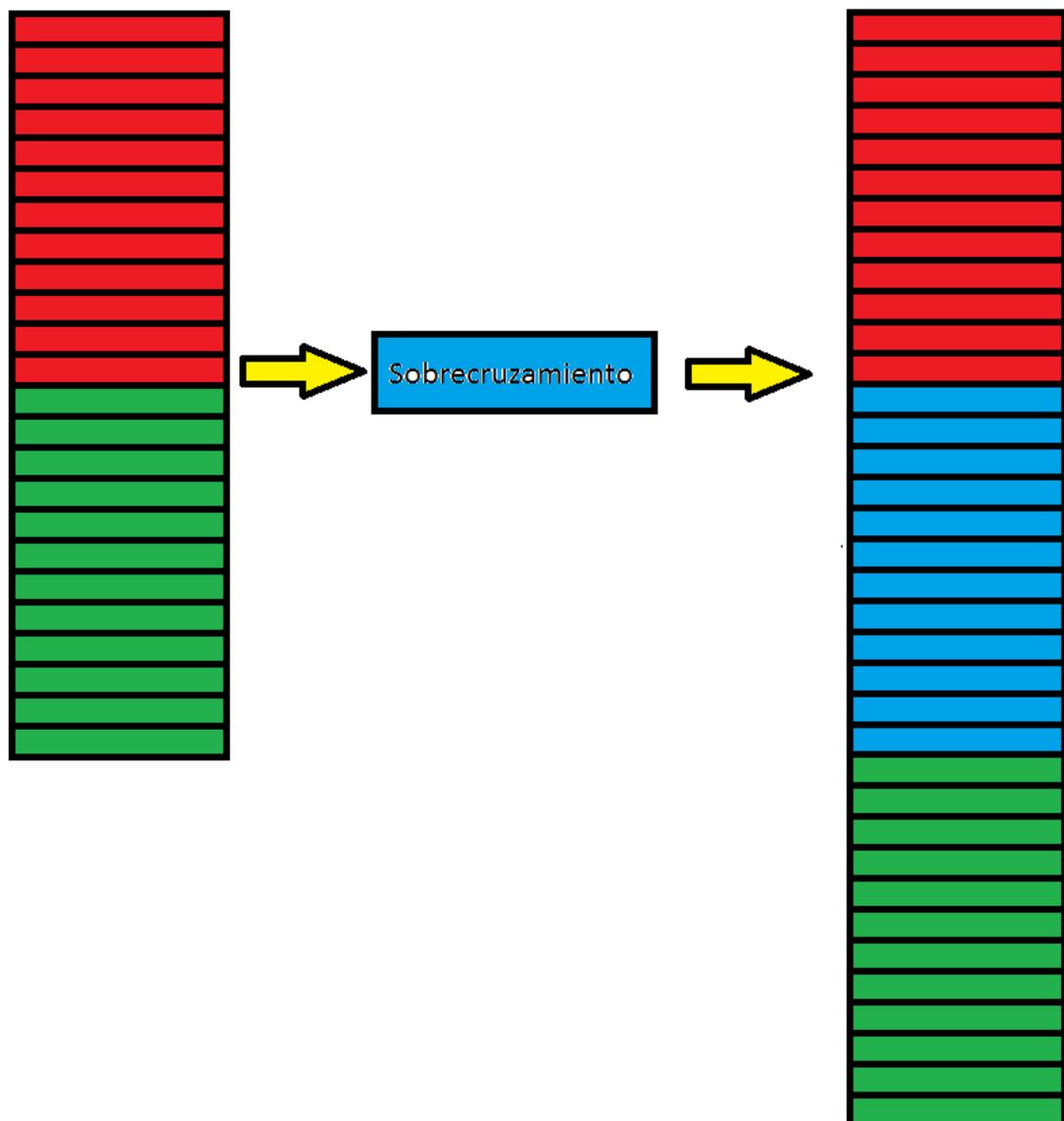
- **Selección:** Se seleccionan de alguna forma diferentes individuos de la población para aplicar operadores de sobre-cruzamiento y mutación.

En este diagrama se muestra el funcionamiento del operador de selección, en este caso torneo. En rojo la población inicial y en verde los individuos seleccionados. Después de aplicar el operador de selección la población crece el mismo número de individuos que se seleccionan.



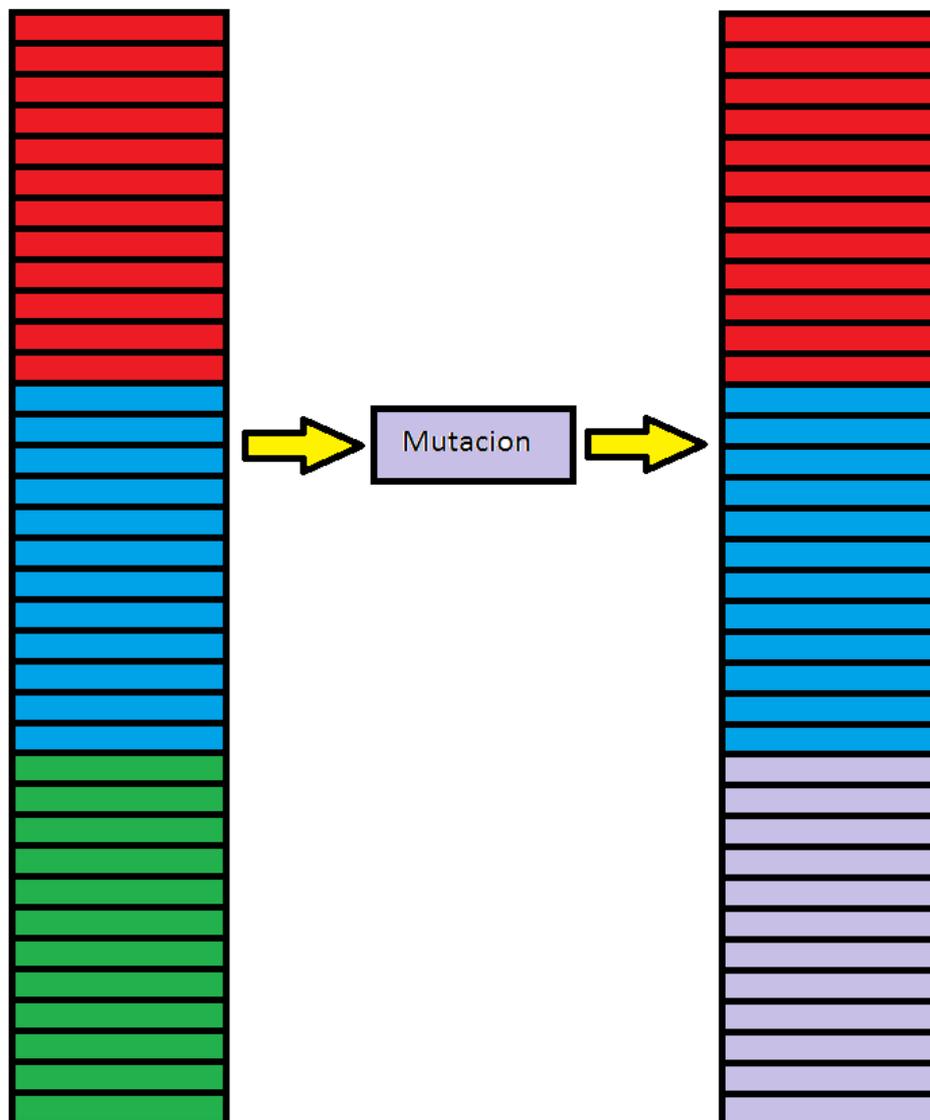
- **Sobre-cruzamiento:** Se seleccionan pares de individuos para generar soluciones a partir de la combinación de información entre un individuo y otro. Los hijos generados por este operador se almacenan al final de la población. El número de individuos que sobre cruzan, como lo hacen y en qué porcentaje puede ser variable.

En el siguiente diagrama se muestra el resultado de aplicar el operador de sobre-cruzamiento. En rojo la población inicial, en verde la población seleccionada y en azul la población después de aplicar el operador de sobre-cruzamiento.



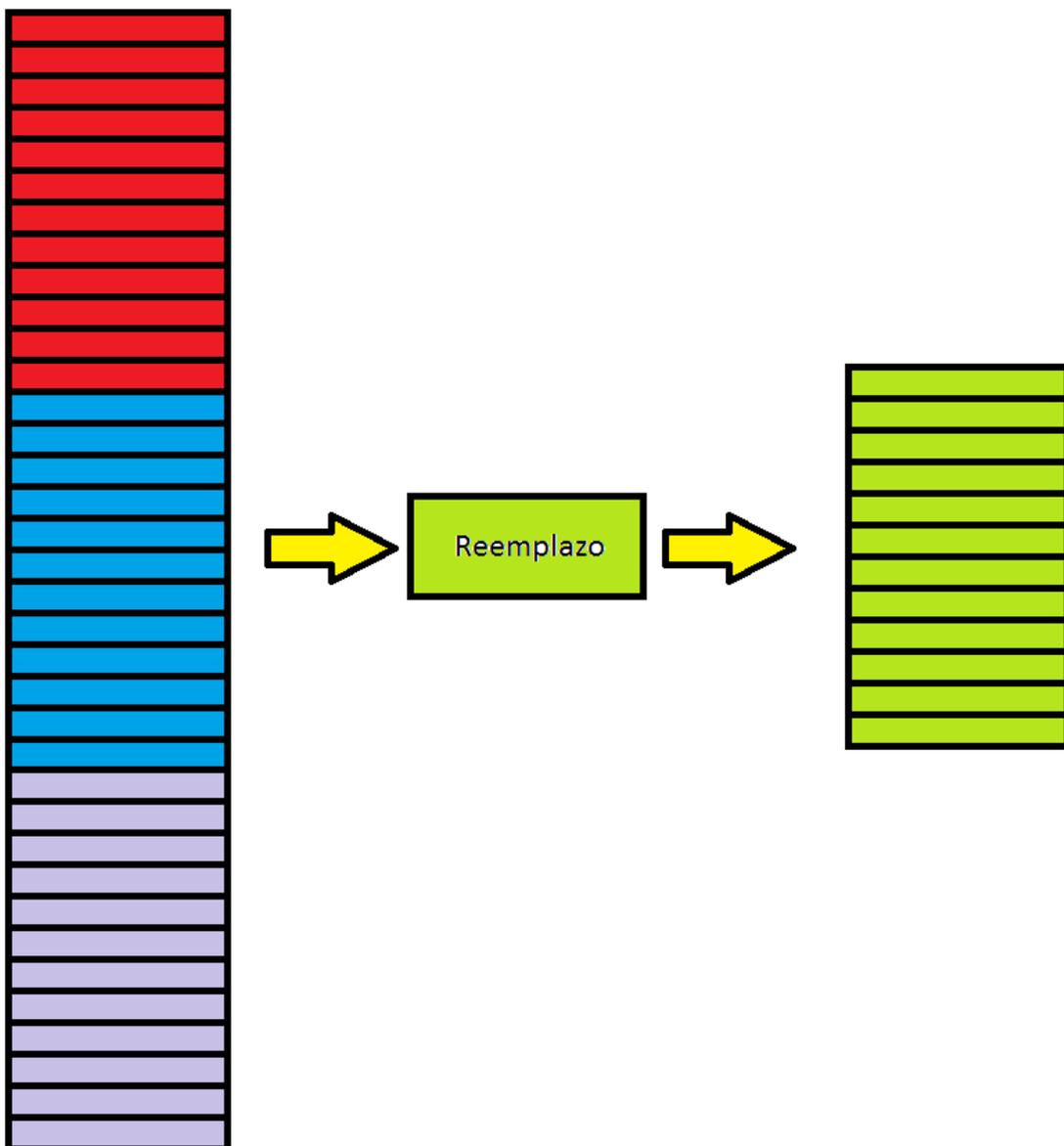
- **Mutación:** Se seleccionan individuos para mutar parte de su información y generar información nueva. El porcentaje de individuos que mutan, como mutan y en qué porcentaje sobre la información que contienen puede ser variable.

En el siguiente diagrama se muestra como quedaría la población después de aplicar el operador de mutación. En rojo la población inicial, en verde la población seleccionada, en azul la población sobre cruzada y en lila la población mutada.



- **Evaluación:** Se evalúa cada individuo de la población y se le da un valor resultante que representa cuanto de buena es esa solución.
- **Reemplazo:** Se seleccionan los mejores individuos de la población después de ser evaluada, el resto de individuos se desechan. El número de individuos que se seleccionan puede ser variable.

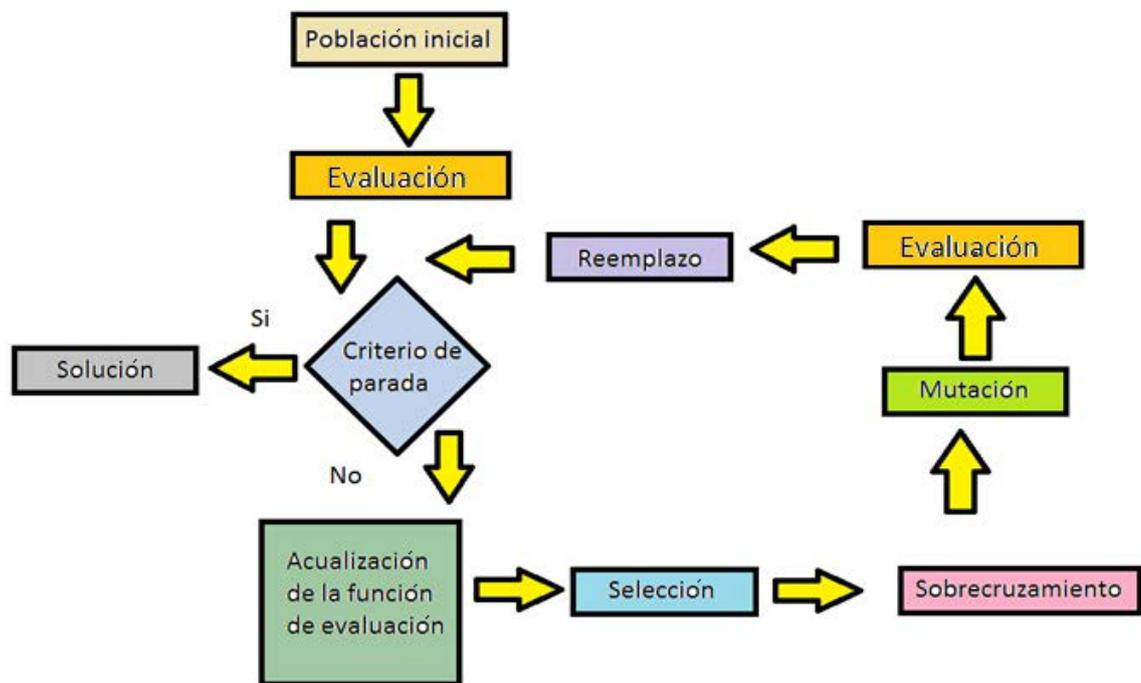
En el siguiente diagrama se muestra la población después de aplicar el operador de reemplazo. En rojo la población inicial, en azul la población sobre-cruzada, en lila la población mutada y en verde la población después de aplicar el operador de reemplazo.



El tamaño de la población resultante ha de ser del mismo tamaño que la población inicial.

El algoritmo para cuando se alcanza un criterio de parada fijado antes de comenzar con la ejecución del algoritmo. Los criterios de parada pueden ser de lo más diversos, siendo esta elección un problema que aún está por resolver.

En el siguiente diagrama se muestra que pasos da un algoritmo genético antes de alcanzar una solución.



Primero se evalúa la población, a continuación se comprueba que se cumple el criterio de parada. En caso negativo la población se selecciona, se sobre-cruza, se muta, se evalúa y se reemplaza. Esta serie de pasos se repite hasta que se cumple con el criterio de parada.



3.3 Modificaciones al CGA

En esta sección se detallaran los cambios realizados sobre el Algoritmo Genético Canónico (sus siglas en ingles CGA).

Las modificaciones que se han realizado sobre el CGA, han sido realizadas para adecuar el algoritmo al problema que se trata en este documento.

Así mismo se explicara en este apartado la función de evaluación que se está usando.



3.3.1 Codificación

La codificación que se ha utilizado, consiste en una estructura n-dimensional. Cada dimensión representa un parámetro que representa un estado concreto durante una partida.

En el apartado de factores importantes se describen una serie de aspectos a tener en cuenta durante una mano de póker.

Para representar a un individuo se han utilizado esos factores. Cada uno de los individuos tiene la capacidad de reaccionar en una situación dada, por lo tanto la representación que se ha utilizado para el espacio de búsqueda es cerrada, puesto que están cubiertas todas las situaciones posibles (dada la representación del espacio de búsqueda).

Los factores que se han utilizado para representar el cromosoma son:

- **La combinación:** La combinación de cartas que tiene el jugador en un momento dado. La combinación puede ser desde carta alta, hasta escalera de color.
- **Fase en la que se encuentra:** La fase en la que se encuentra el jugador en un momento dado. Las fases en las que se puede encontrar³ son: flop, turn y river.
- **El numero de subidas de la fase anterior:** El numero de subidas que ha habido en la fase directamente anterior a la que se encuentra el jugador.
- **La posición del jugador con respecto a la posición de dealer:** La posición⁴ con respecto al dealer que ocupa un jugador en un momento dado.
- **El numero de subidas de la fase actual:** El numero de fichas que ha depositado el jugador que más fichas ha apostado.

³ No se ha tenido en cuenta el preflop a la hora de codificar el cromosoma, ya que como se ha visto antes, se han utilizado tablas pre programadas para la toma de decisiones durante esta fase.

⁴ Las posiciones están descritas en el apartado "2.6 Posiciones".



- **La rentabilidad del bote:** La relación entre el número de fichas que hay que depositar en una acción, entre el bote total que hay en la mesa.
- **Una simulación de la mano:** El resultado de la simulación de la mano que está jugando en un momento dado.

En cada una de las posiciones representadas hay un gen que determina la acción a tomar en esa situación dada.

El gen consta de dos valores enteros, comprendidos entre el rango (-100, 100). Los dos valores se inicializan con un valor aleatorio.

Cuando se da una situación de juego y tiene que tomar una decisión el gen correspondiente devuelve un valor, este valor representa la jugada que va a realizar: subir, igualar o tirarse.

El resultado se calcula de la siguiente manera:

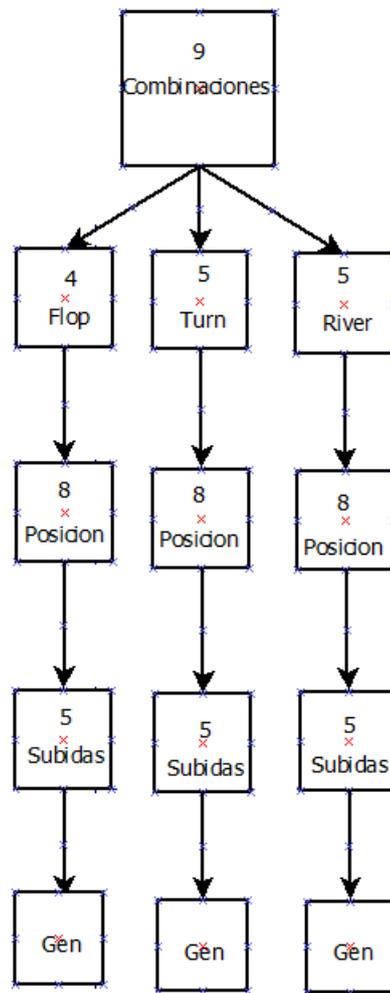
- Cada gen tiene un u_0 y un u_1 .
- La decisión la toma en función del peso de la mano (simulación) y la rentabilidad del bote $d = \text{peso} \cdot \text{rentabilidad del bote}$.
- La decisión sigue la función:

Si $d > u_0$ sube.

Si $d < u_0$ y $d > u_1$ iguala.

Si no se tira.

La figura siguiente muestra el diagrama de la representación de cada individuo.



El número de genes que tiene un individuo es 5040.



3.3.2 Sistema de evaluación co-evolutivo

Para la generación del conjunto de entrenadores que luego utilizaremos para generar al jugador final, se ha empleado un sistema de co-evolución (Seredynski et al. 2003).

Este sistema de co-evolución se emplea debido a la carencia de jugadores con una calidad conocida. La utilización de jugadores aleatorios queda descartada, debido a que no garantiza de ofrecer suficiente dificultad a cada entrenador que se quiere generar. Por otro lado el uso de jugadores pre-programados, tampoco es viable, ya que si se quiere generar un conjunto de entrenadores programados manualmente, primero, si es suficientemente bueno, no necesitaríamos la utilización de técnicas de aprendizaje automático, y segundo llevaría demasiado tiempo generarlo.

El sistema co-evolutivo, garantiza hasta cierto punto la dificultad de los jugadores contra los que juega el individuo que se está entrenando a lo largo de la ejecución del algoritmo, es un sistema que ofrece jugadores diferentes y es poco costoso generarlos.

3.3.3 Conjunto de entrenadores

Para el entrenamiento de los jugadores se ha de utilizar un conjunto de jugadores para que los jugadores puedan medir su rendimiento.

Para ello se utiliza un conjunto de entrenadores. Este conjunto consiste en un conjunto de individuos fijo, con el que se evalúa a toda la población.

Este conjunto, como se verá más adelante, se va modificando a lo largo de la ejecución del algoritmo. Se recurre a este sistema, para evitar la especialización del cromosoma y evitar una convergencia prematura. Otro motivo por el que se recurre a este sistema co-evolutivo, es porque no se dispone de un conjunto de jugadores de calidad, y esto es fundamental si se quiere entrenar a un jugador, ya que se necesita a un entrenador que provoque la evolución del jugador, mediante un juego si no superior, al menos igualado.

3.3.4 Función de evaluación

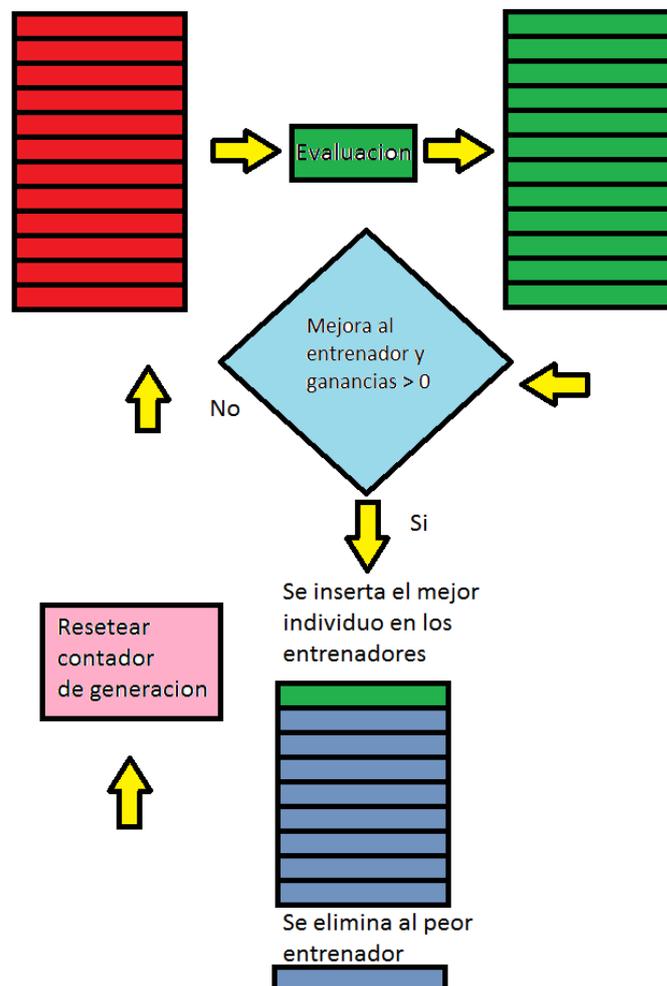
En este apartado se va a describir el funcionamiento de la función de evaluación que se ha empleado para el entrenamiento de los jugadores.

Se trata de una función con un mecanismo de actualización co-evolutivo. Este funcionamiento consiste en ir modificando el conjunto de entrenadores por individuos nuevos.

El funcionamiento es el siguiente:

Se compara el mejor entrenador con el mejor individuo. Si mejora el resultado y las ganancias son mayores a cero se elimina del conjunto de entrenadores al que peor resultado haya obtenido. A continuación se añade al mejor individuo de la población del conjunto de entrenadores.

En el diagrama siguiente se muestra el funcionamiento de la función de evaluación.





El funcionamiento de la función de evaluación es el siguiente: en el caso de que el mejor individuo de la población consiga ganancias mayores a cero y obtenga mejor puntuación que el entrenador que mejor entrenador, inserta al mejor jugador de la población y elimina al entrenador que peor resultado haya obtenido.

De esta manera, la función de evaluación, co-evolucionan junto con la población para que esta se entrene siempre con una diferencia de rendimiento constante.



3.3.5 Operadores

En esta sección se detallaran las modificaciones que se han realizado sobre los operadores de mutación y sobre cruzamiento.

Se han implementado cuatro operadores diferentes para sobre cruzamiento y uno para mutación. Los nuevos operadores modifican información de dos maneras: utilizando información general, modificando aleatoriamente cualquier región del cromosoma y utilizando información usada, es decir, modificando aquellas regiones que han sido usadas durante la función de evaluación.

Durante la función de evaluación se marcan aquellos cromosomas que son utilizados para la toma de decisiones. Esto se ha implementado de esta forma porque una parte importante del cromosoma no se usa por las decisiones tomadas durante la función de evaluación. Por lo tanto queremos focalizar los cambios en la parte usada del cromosoma y así provocar la posibilidad de que se usen partes que no han sido usadas. La relación entre las partes usadas están relacionadas con el concepto de epítasis entre las distintas partes del cromosoma, y por esto se quiere controlar de manera directa (Rochet et al., 1996).

En cada generación se selecciona únicamente un operador de sobre cruzamiento, por tanto, un individuo como mucho puede sobre cruzarse y mutar una única vez por generación.

Cada individuo tiene una probabilidad de sobre cruzarse o mutar entre cero y cien. Cada generación se comprueba la posibilidad de que un individuo se sobre cruce o mute.

Como se verá a continuación, algunos de los operadores dan mayor prioridad a modificar aquellas partes del cromosoma que han sido utilizadas durante la función de evaluación. Se ha decidido utilizar este sistema en los operadores, para forzar que los cromosomas utilicen aquellas partes que no han utilizado antes, y puedan evolucionar un mayor porcentaje del cromosoma.

- **Sobre cruzamiento:** Este operador se encarga de intercambiar información entre pares de individuos. Se seleccionan dos individuos de la población, se les aplica el operador de sobre cruzamiento y se generan dos hijos con la información de los padres intercambiada.

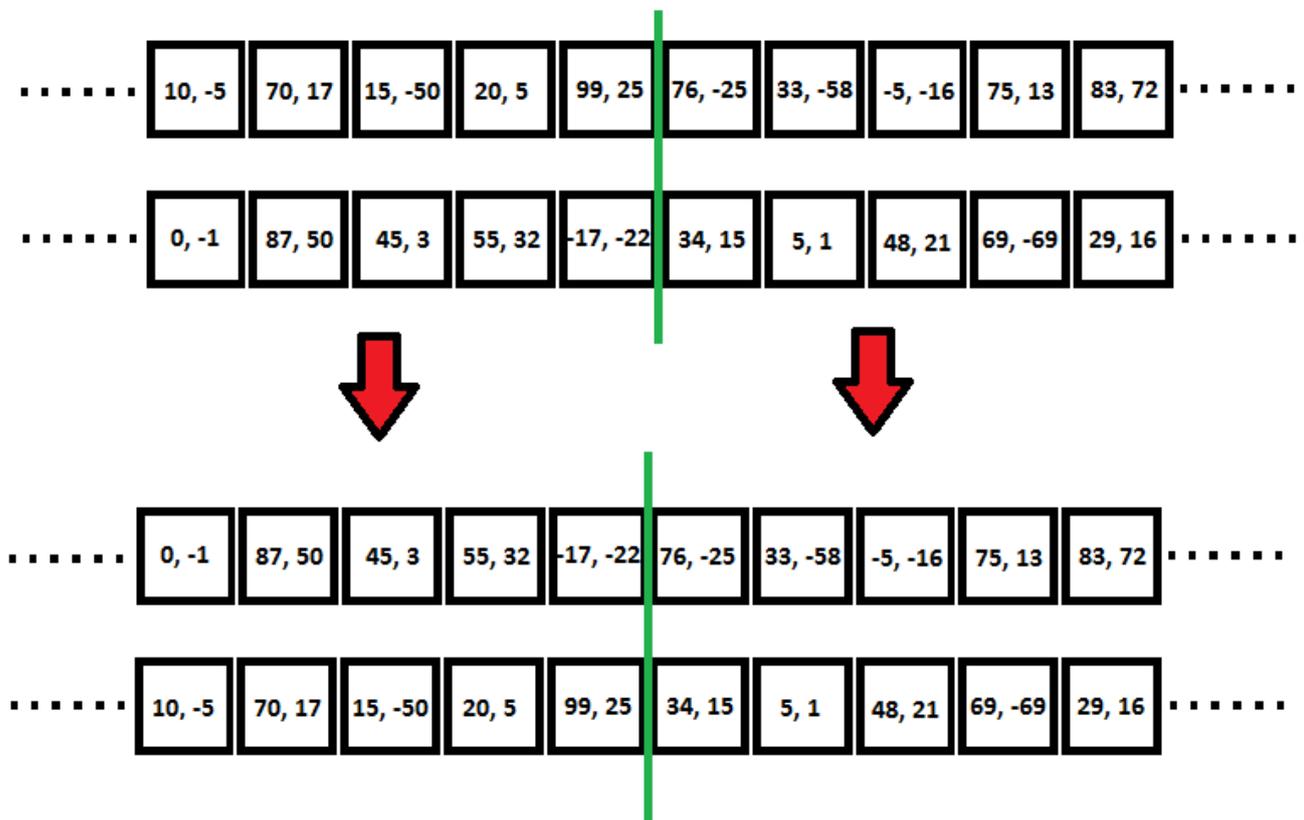
En el sistema que se ha implementado se han realizado cuatro versiones diferentes del operador de sobre cruzamiento, dos teniendo en cuenta los genes usados durante la función de evaluación, y los otros dos sin tenerlo en cuenta.



Así mismo los operadores que se utilizan se diferencian en la forma de elegir los puntos que se seleccionan para intercambiarse. La primera forma es diferenciando dos regiones del individuo e intercambiándolas. La otra forma es seleccionando varios puntos del individuo e intercambiarlos por los respectivos genes del otro individuo.

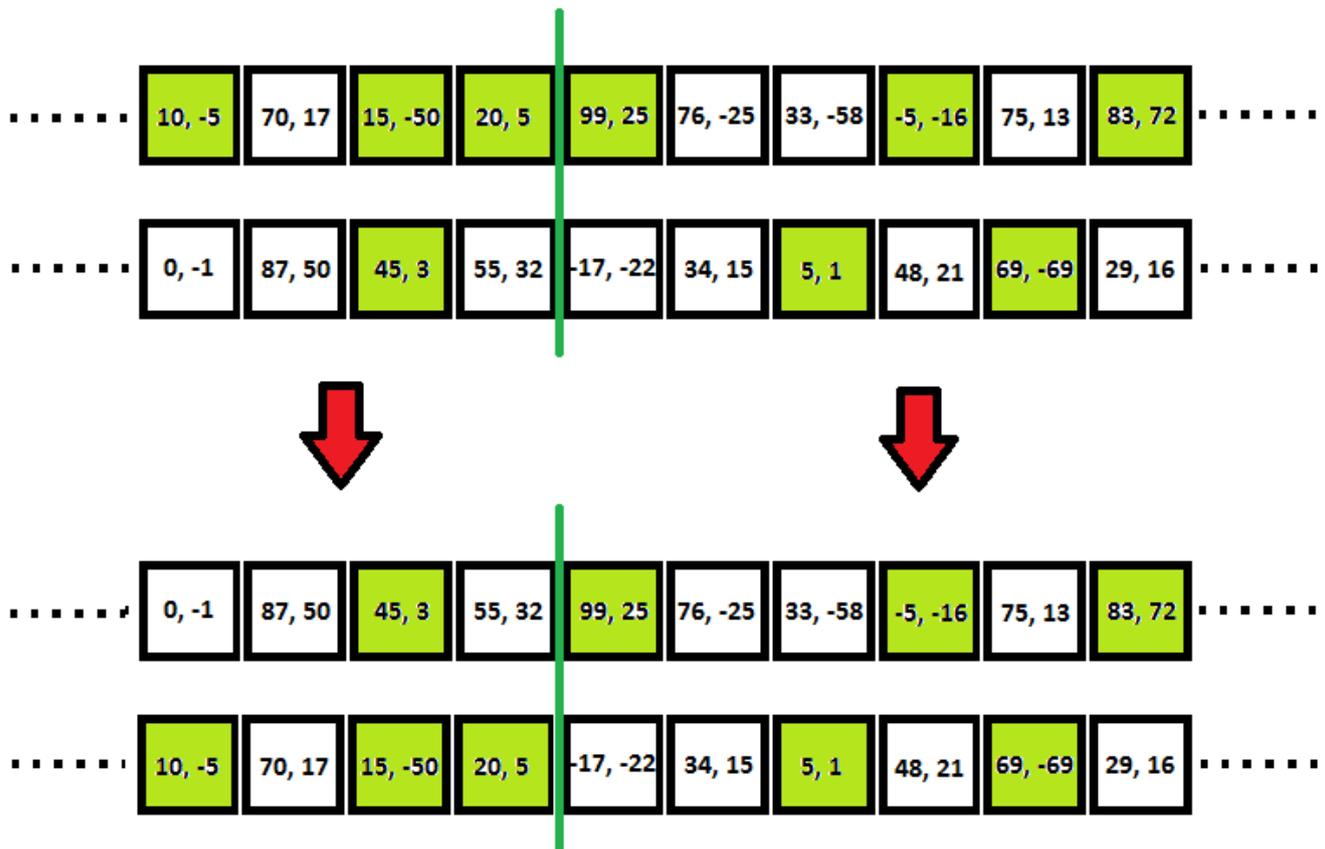
Mitad del individuo: Este operador, selecciona un punto al azar de uno de los dos individuos. Se parten en dos por ese punto a los dos individuos y se generan dos hijos con la primera mitad de uno de los dos padres y la otra mitad del otro de los padres.

El diagrama siguiente muestra un ejemplo de este operador aplicado a dos fragmentos de individuos y los fragmentos de los hijos resultantes. La línea verde indica el punto por donde se van a sobre cruzar los individuos.



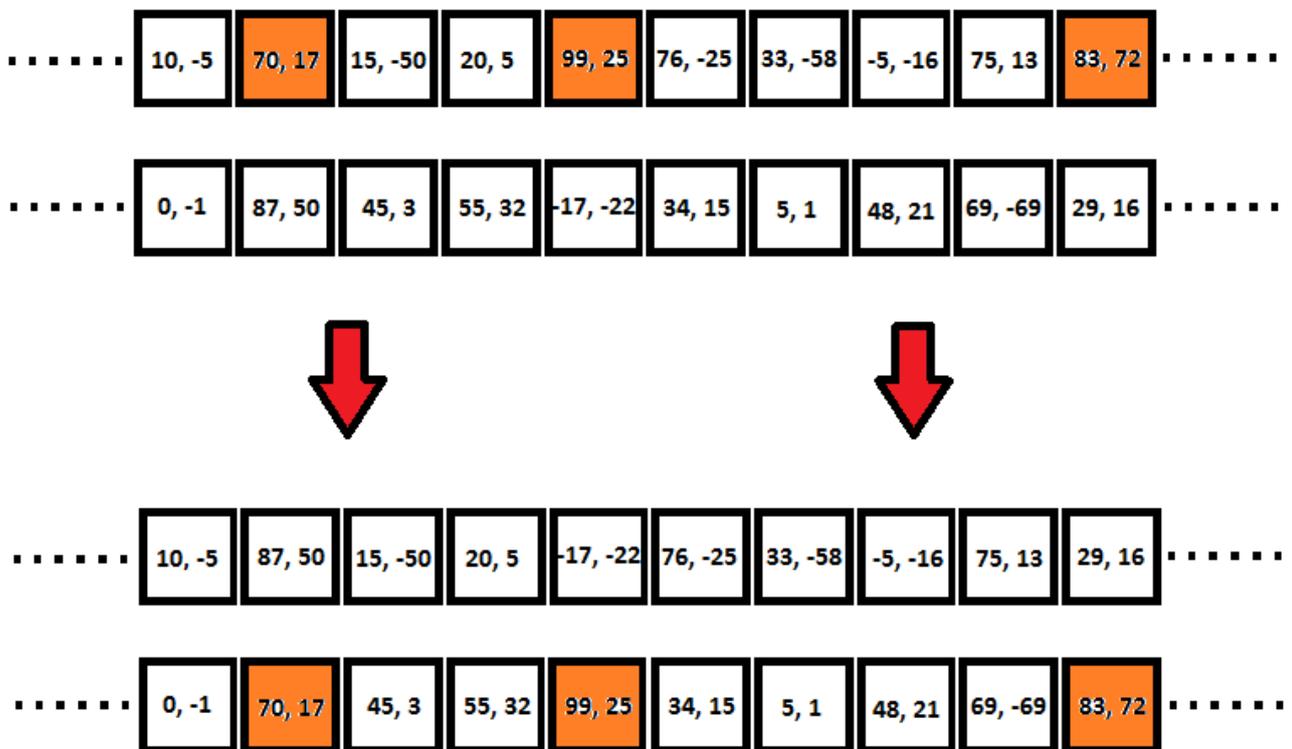
Mitad de los genes usados: Este operador selecciona el punto que representa la mitad de los genes usados por el primer padre. Se parten en dos por ese punto a los dos individuos y se generan dos hijos con la primera mitad de uno de los dos padres y la otra mitad del otro de los padres.

El diagrama siguiente muestra un ejemplo de este operador aplicado a dos fragmentos de individuos y los fragmentos de los hijos resultantes. La línea verde indica el punto por donde se van a sobre cruzar los individuos.



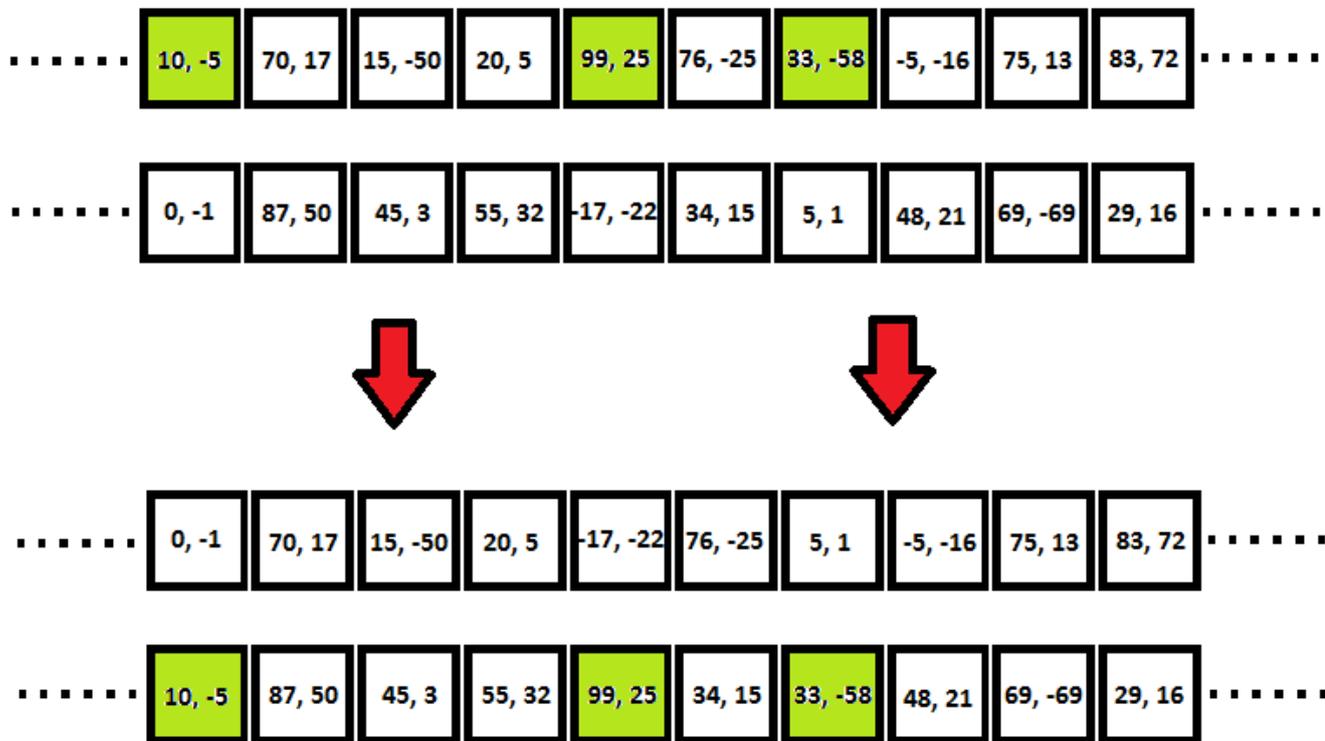
Genes aleatorios: Este operador selecciona varios puntos del individuo de uno de los padres de forma aleatoria. Los padres generan hijos con esos mismos puntos intercambiados.

El diagrama siguiente muestra un ejemplo de este operador aplicado a dos fragmentos de individuos y los fragmentos de los hijos resultantes. Los genes marcados en naranja son los que se han marcado de forma aleatoria para intercambiarse.



Genes usados: Este operador selecciona aleatoriamente varios puntos de uno de los padres que hayan sido usados durante la función de evaluación. Los padres generan hijos con esos mismos puntos intercambiados.

El diagrama siguiente muestra un ejemplo de este operador aplicado a dos fragmentos de individuos y los fragmentos de los hijos resultantes. Los genes en verde son aquellos genes usados seleccionados para sobre cruzarse.

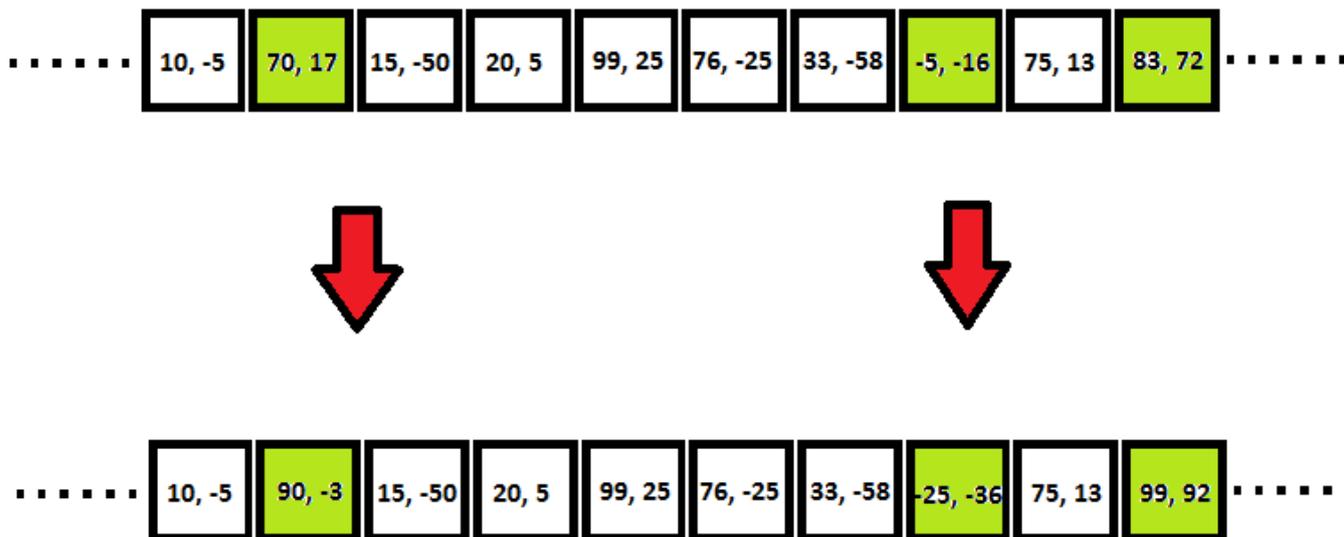


- **Mutación:** Este operador modifica la información contenida en los genes de un individuo.

Los genes se modifican en función de la varianza que se indique por parámetros. La modificación consiste en variar el valor de los dos umbrales más/menos el valor de la varianza por tamaño del rango en el que están comprendidos los umbrales (-100,100).

Se eligen varios genes usados durante la función de evaluación de forma aleatoria, estos genes son los que se mutan.

El siguiente diagrama muestra un individuo sobre el que se ha aplicado el operador de mutación. Los genes marcados en verde son los genes usados que se han seleccionado para mutar.



Varianza = 0.1



4. Implementación

En esta sección se describirá aspectos sobre la implementación que no son propios de un algoritmo genético, pero que se han implementado sobre el sistema tanto para el modelo del problema como optimizaciones de código para aligerar la ejecución y que esta invierta menos tiempo en terminar.

4.1 Introducción

En este apartado se describirá el funcionamiento del simulador que el sistema emplea para la ponderación de las manos en función de las probabilidades de ser la mano ganadora. Se describirá el mecanismo de generación del conjunto de manos que se utiliza para entrenar a los jugadores. Así mismo se explicarán funcionalidades añadidas al sistema para facilitar la evaluación de los jugadores y la extracción de datos sobre los mismos.

También se explicarán las optimizaciones implementadas sobre el sistema, en qué consisten y la justificación de su uso.



4.2 Simulador

Para que el jugador pueda apreciar el valor de una mano, se ha de implementar algún sistema que pondere cada mano en función de las cartas que tiene a la vista en cada fase y de las cartas que aún quedan por salir.

Para la ponderación de las manos se ha implementado un simulador, que simula las posibilidades de ganar que tiene una mano.

El simulador se aplica a en las fases de flop, turn y river. Durante cada una de las fases el simulador fija las cartas que están descubiertas y genera aleatoriamente las manos del resto de jugadores y de las cartas que todavía han de salir en fases posteriores. Cada ronda de simulación, el simulador analiza las combinaciones de cada jugador y determina quién gana, empata o pierde esa mano. Este proceso lo repite un número de veces, almacenando las veces que pierde o las veces que empata o gana.

El resultado del simulador es un valor real comprendido entre 0 y 1, siendo 0 ninguna posibilidad de ganar y 1 que el jugador tiene la mejor mano posible dadas las cartas que hay en la mesa. Por ejemplo, si un jugador tiene una combinación muy alta (un full), cada vez que el simulador, propone cartas nuevas para probar la potencia de la combinación (las cartas que el jugador desconoce se las inventa, sin repetir ninguna), es muy poco probable que aparezca una combinación que sea más potente (solo ganan al full un full más alto, un póker y una escalera de color), por tanto la mayor parte de las iteraciones de simulación saldrá ganadora, dando un peso más cercano a 1. En cambio con una combinación baja (una pareja) es muy probable que otro jugador pueda tener una combinación más alta (además de una pareja más alta, ganan otras siete combinaciones⁵) por lo tanto, en este caso en la mayoría de las rondas de simulación perderá la mano frente a combinaciones más potentes.

⁵ Las combinaciones están descritas en el apartado "2.3.1 Combinaciones".



4.3 Definición de mano

Para el entrenamiento de los jugadores mediante el uso del algoritmo implementado, primero se ha definir lo que dentro del algoritmo llamamos mano.

Una mano en el póker es una ronda de juego, es decir, desde que se reparten las manos hasta que alguien se lleva el bote.

En el caso del algoritmo se ha codificado una estructura que llamamos mano. Esta mano consiste en un juego de 25 cartas, 5 para las cartas de la mano y 20 repartidas entre los 10 jugadores.

Después de que se juegue una mano (mano de póker), se rotan las cartas una posición, dejando las de la mesa fijas. De esta manera cada jugador juega todas las combinaciones disponibles. Cada 10 rotaciones de manos, se rotan a los jugadores una posición. Esto se hace para que cada jugador, además de jugar cada mano disponible, también juegue cada combinación en cada posición.

Por tanto cada mano (mano que estamos definiendo) se utiliza para lo que en el juego del póker se jueguen 100 manos.

Cada mano contiene información relativa a cada combinación de cartas. Esta información representa el valor de la combinación de cada carta y el valor de la ponderación de la mano mediante el simulador implementado.

4.4 Generador de manos

Para el entrenamiento de los jugadores se utiliza un conjunto fijo de manos que se utiliza durante toda la ejecución. Este conjunto de manos ha de ser suficientemente grande como para cubrir la mayor parte del cromosoma posible.

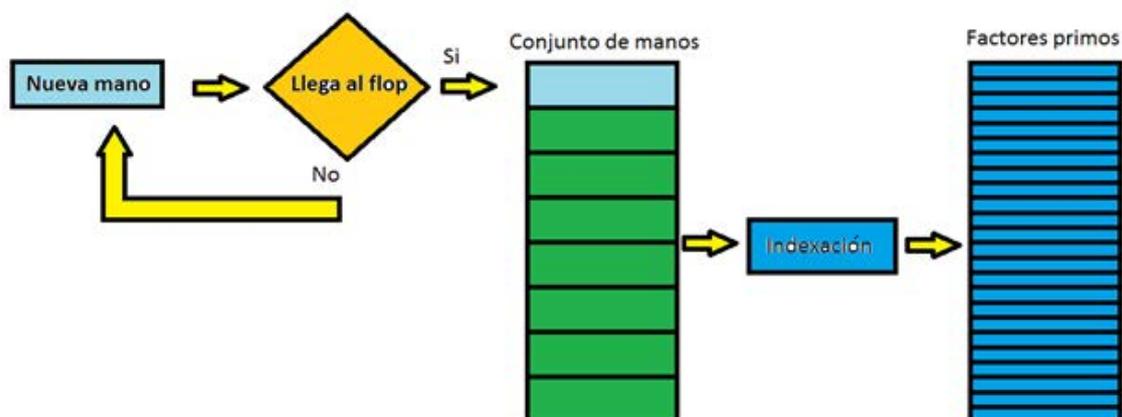
Para generar el conjunto de manos se ha implementado un programa que genera el conjunto de manos siguiendo una serie de premisas con el fin de optimizar el uso de las manos y la inclusión de una serie de información que servirá para la optimización del sistema una vez se ejecute.

El sistema generador de manos, genera un número de manos predefinido. Cada vez que genera una mano comprueba que se alcance la fase de flop al menos en un caso (puesto que lo que se está evolucionando son las fases a partir del flop, sería inútil una mano en la que en ningún caso llega al flop).

En el caso de que se alcance la fase de flop, esta mano se guarda en el conjunto de manos.

Una vez el conjunto tiene el número de manos predefinido, se calcula el peso de cada combinación, es decir un "score" que tiene cada combinación. Estos "scores" se guardan junto con un número que es el resultante de multiplicar un número primo asociado a cada carta involucrada en la combinación. En apartados posteriores se explicará de forma detallada la optimización basada en factores primos.

El diagrama siguiente muestra el funcionamiento del sistema generador de manos.





4.5 Optimizaciones

En este apartado se describirán las optimizaciones que se han llevado a cabo para aligerar la ejecución del algoritmo.

Es importante remarcar que sin las optimizaciones aplicadas el tiempo de ejecución dados los parámetros utilizados hubiera sido inabarcable la ejecución del algoritmo debido a la gran cantidad de tiempo que consume. La cantidad de tiempo que consume, es debido a la cantidad de manos que debe jugar cada jugador durante la función de evaluación. Es necesaria tal cantidad de manos, para que se representen el mayor numero de situaciones posibles y no halla partes de cromosoma sin utilizar.

4.5.1 Modelo de concurrencia

Para la ejecución del algoritmo, se ha implementado un modelo de concurrencia, con el que se divide la carga de la función de evaluación entre un número de hilos predeterminado.

El funcionamiento del modelo de concurrencia, consiste en ejecutar simultáneamente tantas evaluaciones de individuos como hilos hay disponibles. En el momento en el que todos los hilos están ocupados, únicamente se evalúan los individuos que tienen un hilo asignado. Cuando una evaluación termina, se libera un hilo y se le asigna la evaluación de un nuevo individuo, siempre y cuando queden individuos disponibles.

4.5.2 Manos pre-generadas

Esta optimización ya se ha abordado en apartados anteriores, aunque no se ha justificado su implementación.

Se ha optado por este sistema, frente al de calcular todos los datos referentes a las manos durante la ejecución, debido a que el tiempo en obtener esos datos tiene un coste muy elevado, frente al sistema de obtener directamente por parámetros toda la información relacionada con las manos.



4.5.3 Indexación por factores primos

Esta optimización consiste en asignar un score a un número generado multiplicando primos.

Para la implementación de esta optimización, se ha asignado a cada carta de la baraja un número primo diferente. Puesto que la multiplicación de primos genera números unívocos, se pueden indexar las combinaciones por sus factores primos (en un hashmap por ejemplo) y asociar a cada factor primo una combinación.

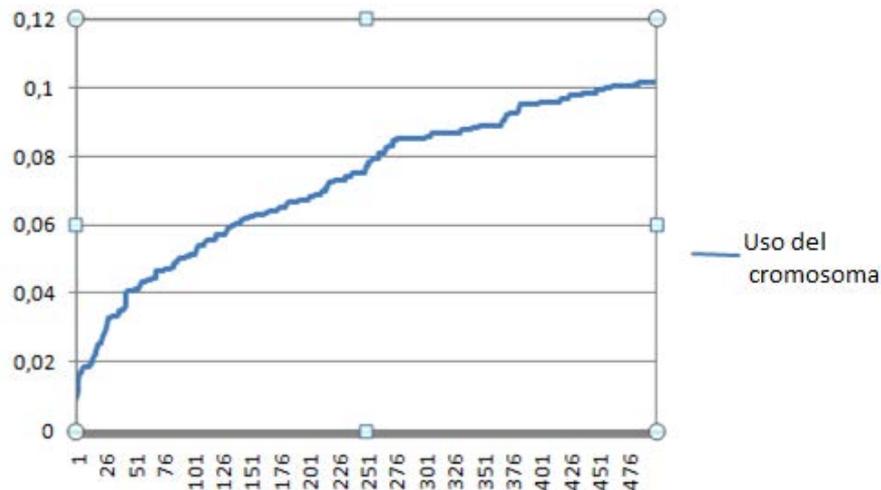
Cada vez que se quiere obtener el "score" de la combinación solo hay que recurrir a obtener la multiplicación de los primos de las cartas que la componen y acceder al "score" que está asociado a ella. Si se implementa con un hashmap, este proceso es tiene un coste lineal.

La alternativa (más costosa) a esta optimización, es ejecutar el algoritmo que decide que combinación se tiene. Este proceso es muy pesado puesto que tiene que ejecutar un algoritmo por combinación en el que se determina que combinación se tiene.

4.6 Coste del uso del cromosoma

El cromosoma representa la solución que se quiere encontrar dentro del espacio de búsqueda que estamos tratando. El uso del cromosoma indica el número de genes que han resultado útiles durante una mano. Un gen ha resultado útil si durante la función de evaluación se ha accedido a su contenido para tomar una acción.

A medida que se juegan más manos el uso del cromosoma va creciendo como se muestra en la siguiente gráfica. El eje de las x representa el número de manos que se juegan, mientras que el eje de las y muestra la cantidad de genes del cromosoma que se están utilizando. El rango de uso del cromosoma está comprendido entre 0 y 1.



Sin embargo como se puede apreciar el uso que se le da al cromosoma (eje de las y) sigue una función logarítmica en función del número de manos, o lo que es lo mismo, el número de manos crece exponencialmente en función del uso del cromosoma.

Esto indica que el coste espacio-temporal del uso del cromosoma crece exponencialmente.

Para hacer un uso exhaustivo del cromosoma en una sola ronda de evaluación, habría que:

- Jugar cada combinación 8 veces.
- Jugar cada flop 10 veces.
- Jugar cada turn 10 veces.

El número de combinaciones es 7200. O lo que es lo mismo, sería jugar 720 manos suponiendo una distribución de combinaciones homogénea. Teniendo en



cuenta la distribución de combinaciones representada en la primera tabla del apartado de combinatoria, habría que jugar 5179648 manos en el mejor caso.

Sin embargo a medida que evoluciona el individuo, va explorando diferentes regiones de su codificación. Por lo que no es necesario utilizar ese número de manos, sino que con que quede representada al menos una escalera de color en el conjunto de manos hemos considerado que es suficiente, además de que con los recursos de los que disponemos, un conjunto de manos mayor haría casi inabarcable el problema, debido al tiempo que se necesitaría.



4.7 Criterios de convergencia

Existen varios criterios de convergencia para este algoritmo, cada uno para un caso diferente. A continuación se detallan los diferentes criterios de convergencia que se han establecido para determinar si es necesario continuar con la ejecución. El algoritmo converge cuando ha perdido su variedad genética y la población entera explora una única solución.

Se puede dar el caso de una convergencia prematura del algoritmo. Esto ocurre cuando el conjunto de manos es demasiado pequeño y no puede mejorar el conjunto de entrenadores. Se establece que ha convergido prematuramente cuando para mejorar un entrenador invierte más número determinado de generaciones en sin conseguirlo.

Para determinar si el algoritmo ha conseguido converger a un sub-óptimo, este se prueba con un conjunto de entrenadores de test. Estos entrenadores no influyen en la evaluación y son fijos, es decir, nunca se modifica.

Cuanto mayor sea el espacio de búsqueda, independientemente de la codificación propuesta, más tiempo tardará en converger, .



4.8 Generación de log para PokerTracker 3

Todas las acciones que los jugadores realizan se almacenan con el fin de evaluar su comportamiento y analizar datos relevantes sobre sus conductas. Por consiguiente, es necesario determinar una estructura de datos que almacene eficientemente toda esa gran cantidad de información con el fin de minimizar su ocupación física y facilitar su recuperación. De la misma manera se habría de implementar un sistema que tratase esa información y la dispusiese de una forma coherente y accesible, para que se pueda utilizar posteriormente para un análisis del entrenamiento.

Realizar una tarea de este tipo es un proceso muy costoso que hubiera involucrado muchas horas únicamente en el sistema de análisis de datos.

Los jugadores de póker humanos que juegan en comunidades on-line (también hay sistemas capaces de analizar manos mediante información obtenida con una cámara), utilizan herramientas de análisis de juego, para evaluar cómo juegan y obtener datos que puedan servir para ayudarles a mejorar su juego. Los jugadores de poker on-line extraen la información del juego mediante un log que genera el propio cliente de la sala de póker y almacena esa información en el ordenador del jugador humano.

Para el análisis de los datos de juego de los jugadores que se están entrenando se ha optado por copiar el sistema de generación de log de un cliente on-line, y utilizar la herramienta PokerTracker 3 para el análisis de datos.

Esta herramienta, permite desde re-jugar manos con una interfaz muy cómoda, visualizar graficas, aplicar cualquier tipo de filtro para estudiar situaciones concretas. Esta herramienta es realmente potente y su uso para el análisis de los individuos que se generan es realmente útil.

Utilizando los conocimientos aprendidos sobre el procesamiento del lenguaje, se analizó el log que genera el cliente de póker on-line "PokerStars". En este análisis se diferenció que partes del log variaban en función de la jugada, que partes son indispensables para que PokerTracker 3 no considere que el log de la mano tiene errores (si PokerTracker 3 detecta un error, esa mano la desecha, y por tanto se pierde esa información), que partes del log no varían y que partes se pueden omitir sin que se produzcan errores.

La generación del log para su posterior análisis, es perfecta, ya que no genera ningún error y ha resultado extremadamente útil a la hora de elaborar estadísticas y seguimientos de los individuos.

Durante la ejecución del algoritmo genético, se genera un log en el que están detalladas todas las acciones de los jugadores, para su posterior importación con la herramienta PokerTracker 3 y su análisis.



5. Resultados

En esta sección se detallarán los resultados obtenidos de las de las ejecuciones realizadas para generar tanto los entrenadores para entrenar al jugador final como la ejecución del entrenamiento del jugador final.

Este apartado expondrá los parámetros utilizados para cada una de las ejecuciones realizadas para la generación de los jugadores. Así mismo se explicará en qué consisten estos experimentos y cuál es la finalidad de cada uno.

Por último se expondrán una serie de gráficas en las que quedan representados los datos más significativos de la ejecución del sistema.



5.1 Parámetros utilizados

Los parámetros escogidos para la ejecución, se han elegido para equilibrar rendimiento del sistema con tiempo de ejecución.

Se han escogido dos tipos de parámetros dependiendo del tipo de jugador que se quiera obtener. Para el entrenamiento de los entrenadores, se han escogido conjuntos de manos más pequeños, puesto el tiempo de ejecución es considerablemente alto (aproximadamente 20 horas con los parámetros escogidos). Para el entrenamiento del jugador final, se ha utilizado un conjunto de manos superior, puesto que el sistema es más ligero y puede jugar más manos en menos tiempo.

Los parámetros que entran en juego a la hora de calcular la complejidad son:

- **Número de manos:** El número de manos que se juega durante la evaluación de la población.
- **Número de generaciones:** El número máximo de generaciones que puede invertir el algoritmo en generar un nuevo entrenador.
- **Número de entrenadores:** El número de entrenadores que ha de generar el algoritmo antes de parar.
- **Tamaño de la población:** El tamaño de la población que se va a utilizar para el entrenamiento.
- **Probabilidad de mutación:** La probabilidad que tiene un individuo de mutar.
- **Veces que muta:** El número de veces que muta el cromosoma en una generación.
- **Probabilidad de sobre cruzamiento:** La probabilidad que tiene un individuo para sobre-cruzarse.
- **Puntos de sobre cruzamiento:** El porcentaje del cromosoma que se sobre-cruza.
- **Número participantes en el torneo:** El número de participantes en el operador de torneo.
- **Puntos que sobre cruza:** El número de puntos que sobre-cruza.
- **Varianza:** La varianza que aplica el operador de mutación.



Estos parámetros afectan a la complejidad del problema, ya que indican el ratio al que se explora el espacio de búsqueda. Sin embargo los parámetros más importantes que son clave para la convergencia del algoritmo son el número de manos (representa un subconjunto del espacio de búsqueda que se va a explorar) y el tamaño de la población (representa número de soluciones disponibles). Cuanto mayor sea el número de manos más tiempo tardará en converger, puesto que el espacio de búsqueda se agranda. Cuanto mayor sea el tamaño de la población mayor será el espacio de búsqueda que se explore y antes encontrará una solución.

Sin embargo en este documento no se abordará el problema de la selección óptima de parámetros ya que ese sería un problema totalmente distinto al que se aborda en este documento.

A continuación se muestra una tabla con los parámetros escogidos para el entrenamiento de los entrenadores y del individuo final.

Parametro	Entrenadores	Jugador final
Generaciones	200	500
Entrenadores	500	N/A
Tamaño de la población	100	100
Numero de manos	200	520
Numero de entrenadores	9	9
Sobrecruzamiento	1	1
Mutación	1	1
Puntos	2520	2520
Veces	504	504
Participantes en el torneo	3	3
Varianza	0,1	0,1



5.2 Prueba de generación de entrenadores

La prueba de generación de entrenadores, consiste en un juego de 9 experimentos en el que cada uno entrena a un entrenador diferente.

Cuando los entrenadores estén listos se utilizaran para el entrenamiento del jugador que se quiere conseguir.

Estas pruebas, debido a la función de evaluación que se está utilizando, es muy pesada.

Los conjuntos de manos utilizados para el entrenamiento de los entrenadores, aunque tienen el mismo tamaño, son diferentes y se han generado con una semilla diferente.

Las semillas utilizadas para generar los conjuntos de manos, la población inicial y para el funcionamiento de los operadores, es la misma dentro de una ejecución correspondiente a un entrenador, pero diferente para cada entrenador.

5.3 Prueba de generación de individuos contra entrenadores

La prueba de generación de individuos a partir de entrenadores, se realiza una vez se tiene el conjunto de entrenadores.

Este conjunto de entrenadores es el que se utiliza para generar a un individuo capaz de tener una capacidad de juego más efectiva.

Este entrenamiento es más ligero puesto que no se ha de utilizar la función co-evolutiva descrita previamente, sino que el conjunto de entrenadores permanece fijo a lo largo de toda la ejecución.

Por este motivo se utiliza un conjunto de manos más grande que el utilizado para el entrenamiento de los entrenadores. Esto le otorga al entrenador una perspectiva más general de juego, puesto que tiene disponible una cantidad de situaciones mayor.



5.4 Análisis de resultados

En esta sección se analizarán los resultados del entrenamiento de los jugadores que se han desarrollado para generar el conjunto de entrenamiento para entrenar al jugador final, así como los resultados del jugador final.



5.4.1 Entrenadores generados

Las graficas que se muestran a continuación, describen el comportamiento de cada entrenador generado.

Se ha generado un conjunto de nueve entrenadores, con los que luego se va a entrenar al individuo final.

Todas las graficas y tablas generadas con el PokerTracker 3 se han filtrado, a únicamente aquellas manos con las que se ha conseguido llegar a la fase del flop. Puesto que la parte del juego que se está evolucionando es a partir del flop, incluir aquellas manos en las que no se ha visto el flop, únicamente añadirían ruido a los resultados.

A continuación se explicara el significado de las tablas y graficas:

- **20 mejores manos:** Esta tabla muestra las 20 manos que ganan más dinero junto con la probabilidad de ganar y las veces que se ha ganado. Las manos que están en este conjunto deberían coincidir con las manos más potentes del preflop.
- **Combinaciones:** Esta tabla muestra las combinaciones que se han jugado, la probabilidad de ganar el dinero total ganado y las veces que ha tenido el jugador esa combinación.
- **Grafica de ganancias en la última generación:** Esta grafica muestra el dinero total ganado a lo largo de la última generación (verde), el dinero ganado enseñando las cartas (azul) y el dinero ganado sin enseñar las cartas (rojo). El eje de las x representa el número mano y el eje de las y representa la cantidad de dinero ganado.
- **Ganancias durante la ejecución:** Esta grafica muestra el dinero ganado por el mejor individuo a lo largo de la ejecución (azul) jugando contra el conjunto de entrenadores. También muestra el dinero ganado por el mejor entrenador (rojo) a lo largo de la ejecución. El eje de las x representa la generación de forma absoluta⁶ y el eje de las y representa la cantidad de dinero ganado.
- **Evolución frente a entrenadores aleatorios:** Esta grafica muestra la evolución de la puntuación del mejor individuo (azul)

⁶ El contador de generación se resetea cada vez que se modifica el conjunto de entrenadores, pero se puede llevar un control del numero de generación total.



compitiendo contra jugadores aleatorios (rojo). El eje de las x representa el número de entrenador y el eje de las y representa la cantidad de dinero ganado.

- **Evolución frente a entrenadores regenerados:** Esta grafica muestra la evolución del mejor individuo (azul) compitiendo contra un conjunto de jugadores generado previamente (rojo). Estos entrenadores se generaron con una versión anterior del sistema implementado. El eje de las x representa el número de entrenador y el eje de las y representa la cantidad de dinero ganado.
- **Coste generacional por entrenador:** Esta grafica muestra el coste generacional de cada entrenador. El eje de las x representa el número de entrenador y el eje de las y representa el numero de generaciones que se han invertido en generar el siguiente entrenador.

Las graficas en las que se describe el comportamiento de la evolución del entrenador (tanto en las que se describen las puntuaciones como el coste generacional), presentan una forma ligeramente sinusoidal.

Este comportamiento se debe a la forma de evolución del conjunto de entrenadores. Si aparece un individuo muy superior al conjunto de entrenadores, este individuo poblara rápidamente el conjunto de entrenadores con un coste generacional muy bajo. Si el conjunto de entrenadores contiene muchos jugadores iguales, baja el dinero que ganan, ya que juegan de la misma forma. Debido a esto el dinero que la mesa se lleva de impuestos (rake) es más alto provocando que cada entrenador nuevo conlleve un coste generacional más alto. Este proceso es cíclico a lo largo de toda la evolución.

A continuación se muestran los resultados obtenidos de la generación de los entrenadores.



5.4.2 Entrenador 0

- 20 mejores manos.

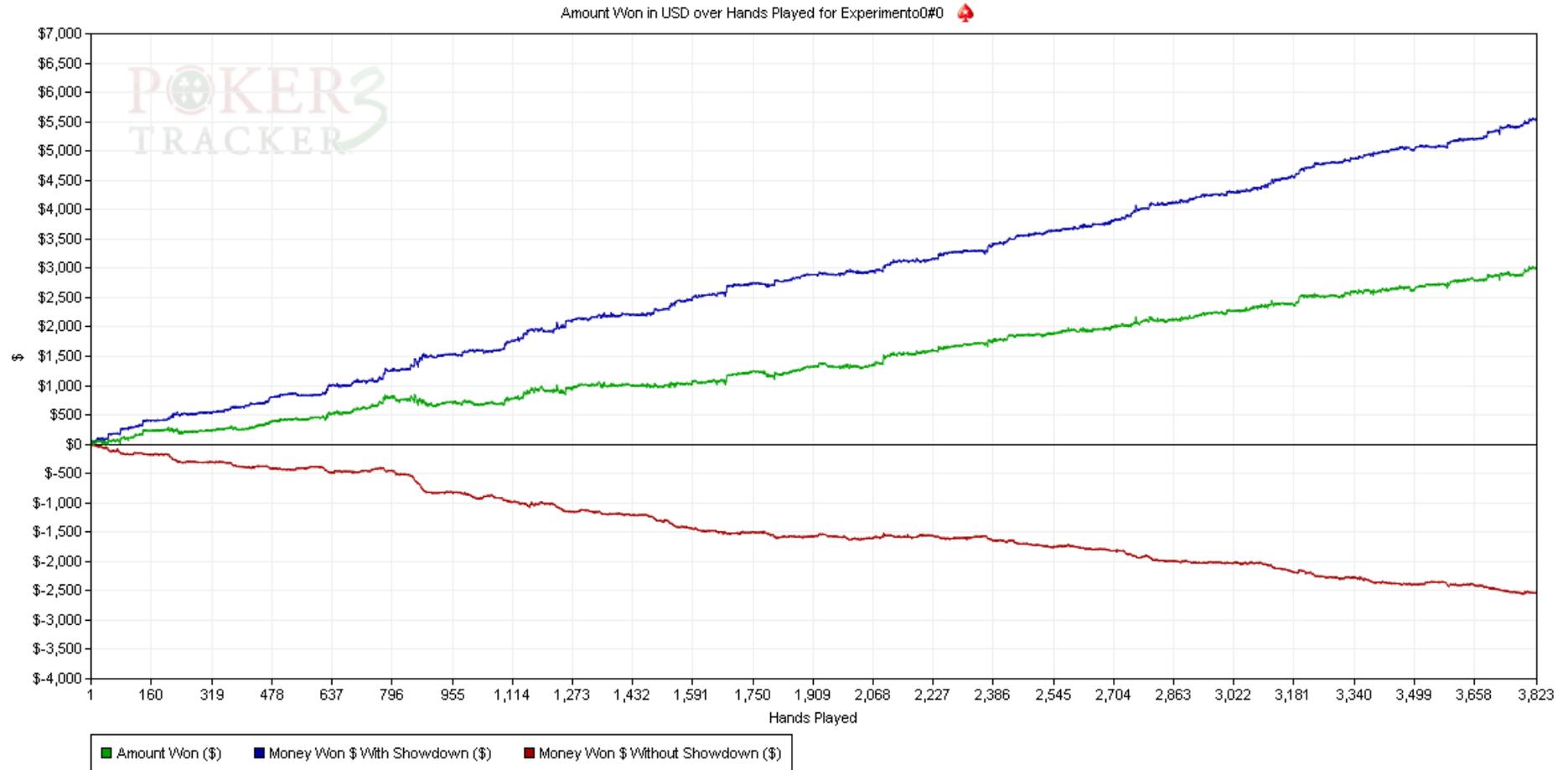
Hand	Times	Win %	Amount Won
AA	67	71.64	\$571.00
AJo	206	50.97	\$536.87
JJ	60	55.00	\$513.25
QQ	69	60.87	\$477.50
ATs	90	54.44	\$423.12
88	118	49.15	\$387.00
99	102	46.08	\$379.06
KK	96	67.71	\$265.50
QJo	91	49.45	\$251.50
KTo	82	48.78	\$228.99
QJs	31	48.39	\$221.66
JTo	73	27.40	\$211.33
AQs	70	42.86	\$202.25
AKs	14	85.71	\$202.25
33	34	32.35	\$195.25
98s	15	80.00	\$186.25
KJo	113	52.21	\$185.33
87o	40	37.50	\$167.00
A9s	29	51.72	\$139.25
K9s	24	54.17	\$133.75

- Combinaciones.

Final Hand	Times	Win %	Amount Won
Straight Flush	1	100.00	\$2.75
Four of a Kind	8	87.50	\$300.00
Full House	120	77.50	\$1,733.61
Flush	73	65.75	\$514.00
Straight	210	85.24	\$2,234.55
Three of a Kind	255	69.80	\$2,339.50
Two Pair	946	48.52	\$1,444.84
One Pair	1,481	32.55	-\$2,851.51
High Card	729	18.24	-\$2,706.75

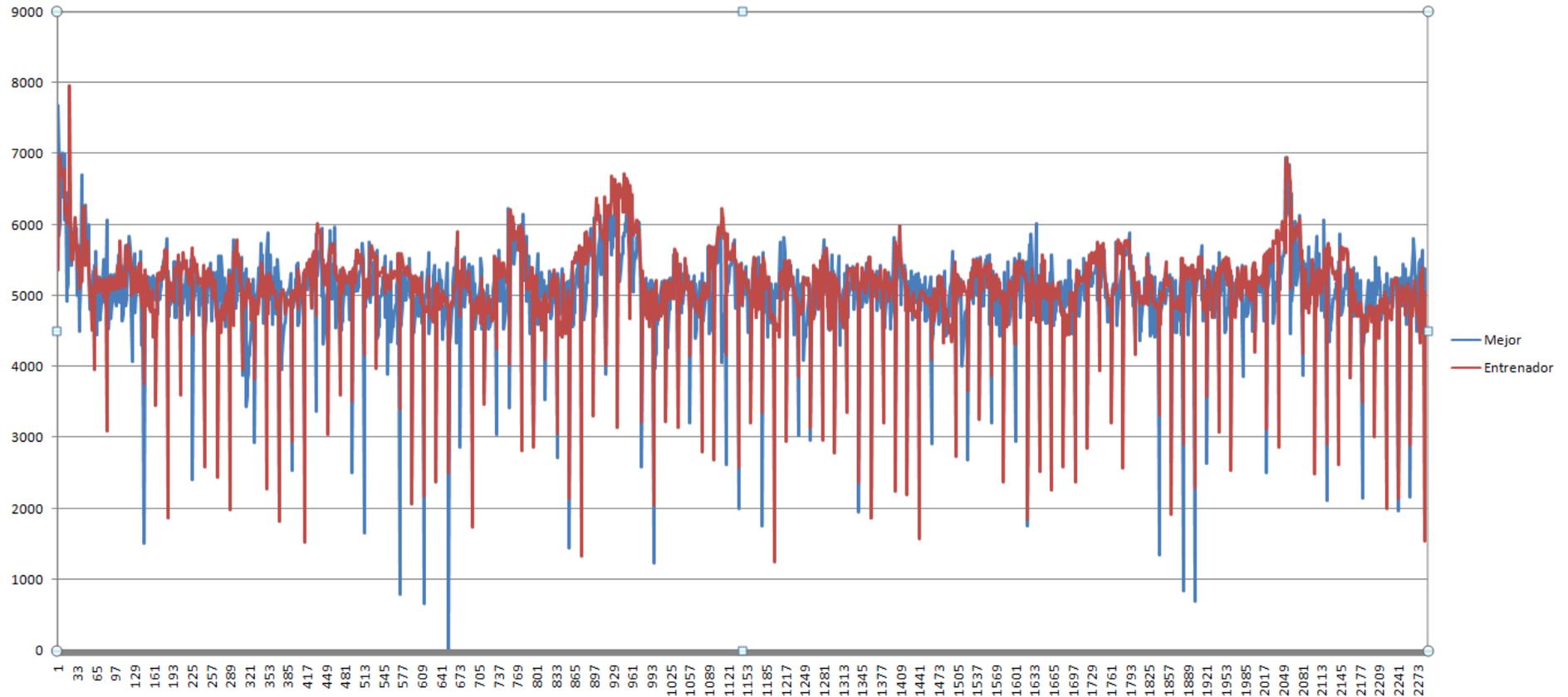


- Grafica de ganancias en la última generación.



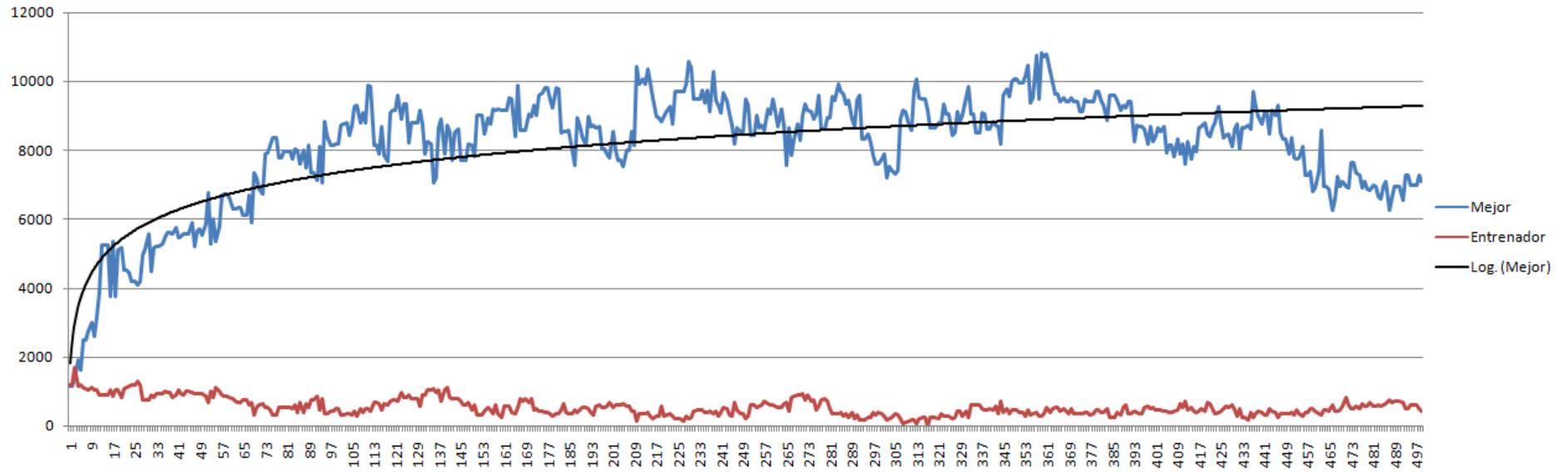


- Gráfica de ganancias durante la ejecución.



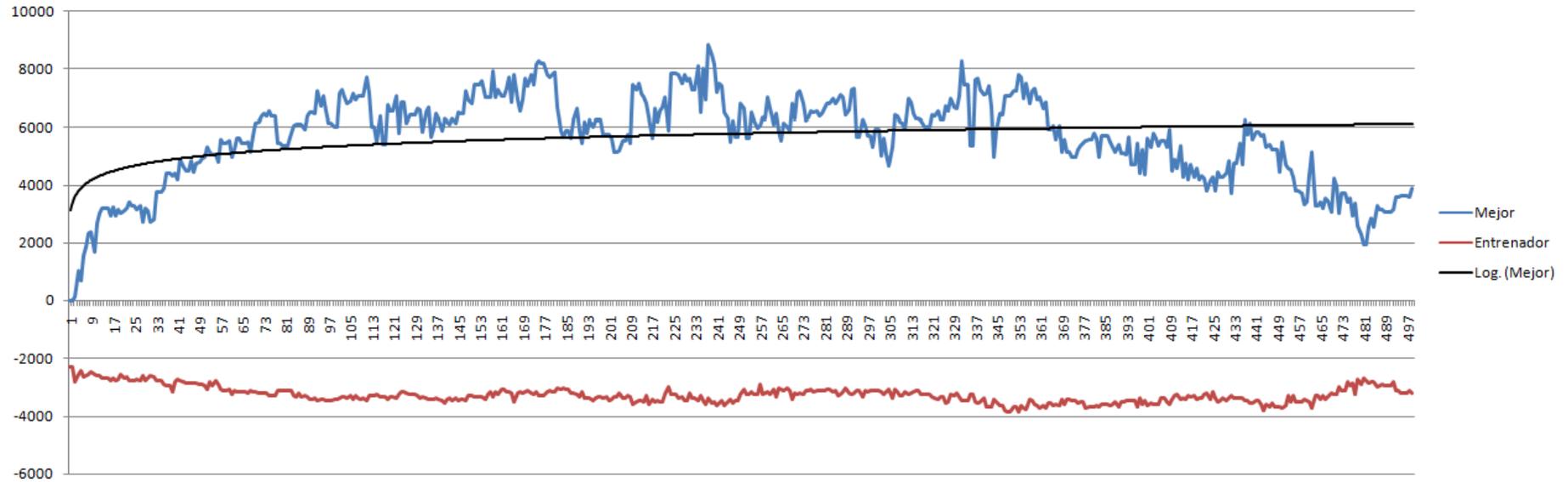


- Grafica de evolución frente a entrenadores aleatorios





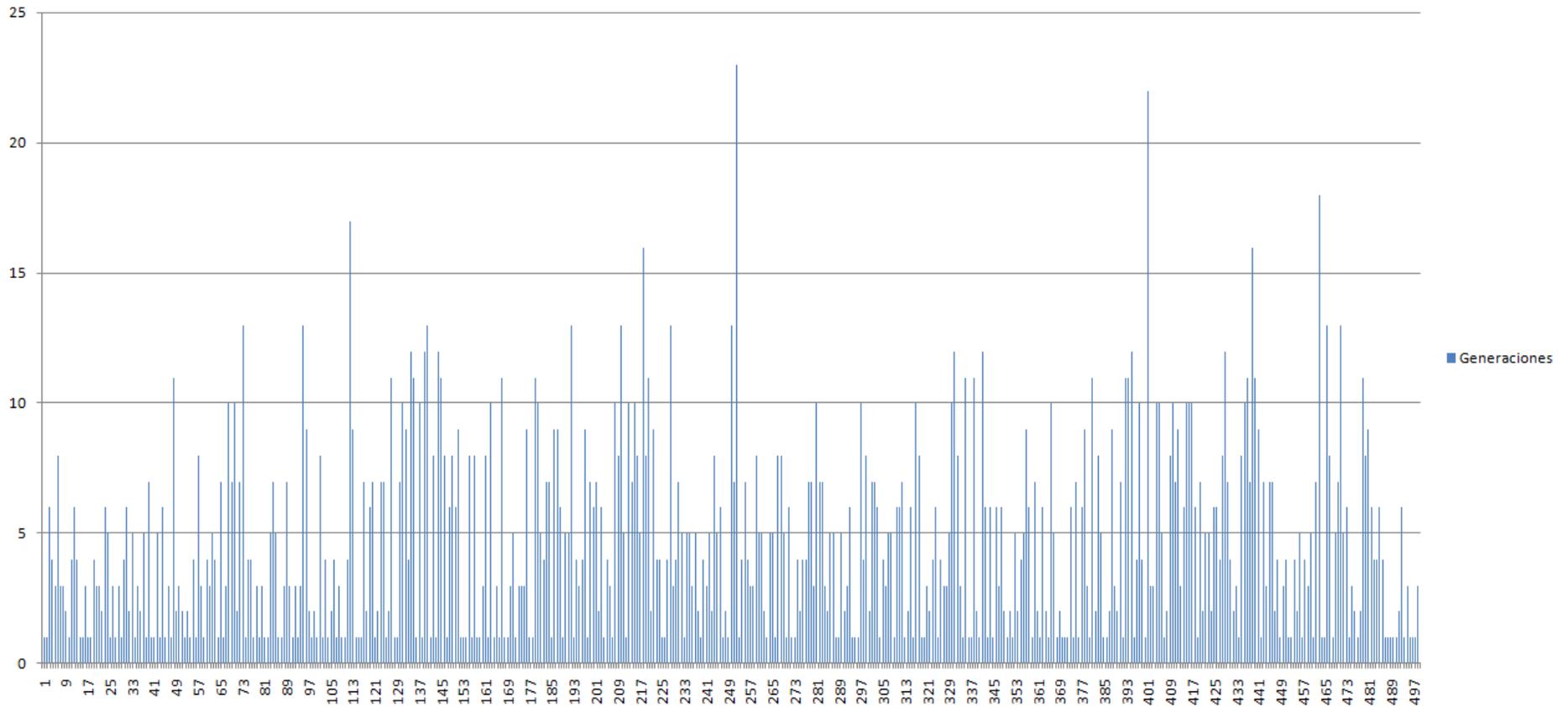
- Grafica de evolución frente a entrenadores pre generados.





- Grafica de coste generacional por entrenador.

Generaciones





5.4.3 Entrenador 1

- 20 mejores manos.

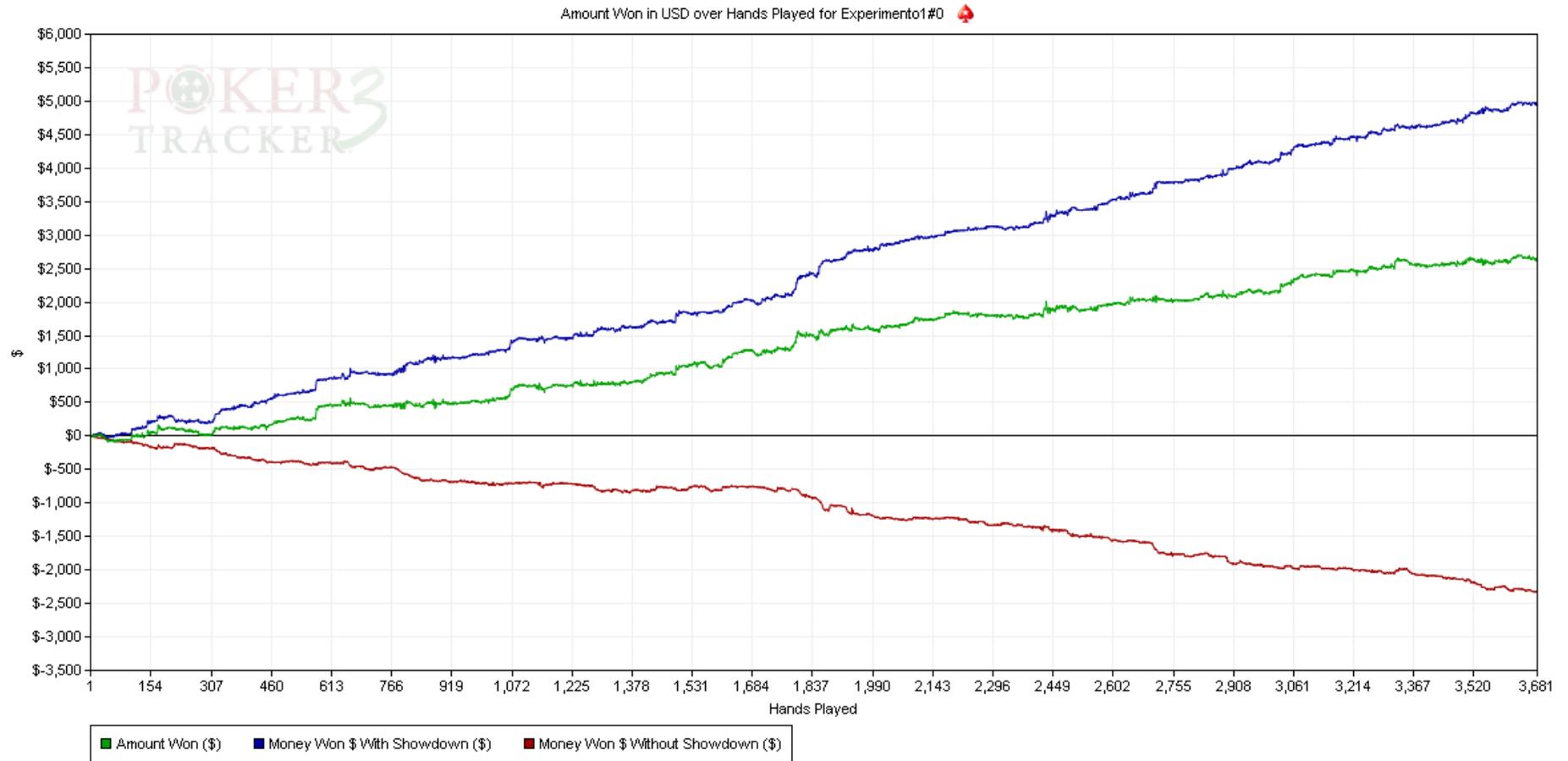
Hand	Times	Win %	Amount Won
TT	87	73.56	\$1,100.75
88	76	67.11	\$863.00
AQo	119	68.07	\$761.60
99	75	50.67	\$703.25
AA	75	58.67	\$646.00
AJo	216	51.85	\$616.42
JJ	53	64.15	\$615.50
AJs	50	72.00	\$459.86
ATs	55	70.91	\$394.37
77	80	55.00	\$382.75
AKo	127	52.76	\$280.11
A9o	66	51.52	\$254.16
KJs	59	55.93	\$229.50
QTo	85	47.06	\$207.50
KJo	143	52.45	\$176.00
T9o	58	43.10	\$148.00
KK	33	57.58	\$141.75
Q9o	35	54.29	\$89.99
65s	18	61.11	\$85.75
J9s	10	60.00	\$57.00

- Combinaciones.

Final Hand	Times	Win %	Amount Won
Full House	151	91.39	\$4,140.75
Flush	103	66.99	\$1,155.25
Straight	156	76.28	\$1,551.87
Three of a Kind	199	69.85	\$1,437.75
Two Pair	866	49.08	\$13.28
One Pair	1,513	34.17	-\$3,032.24
High Card	693	24.96	-\$2,650.51

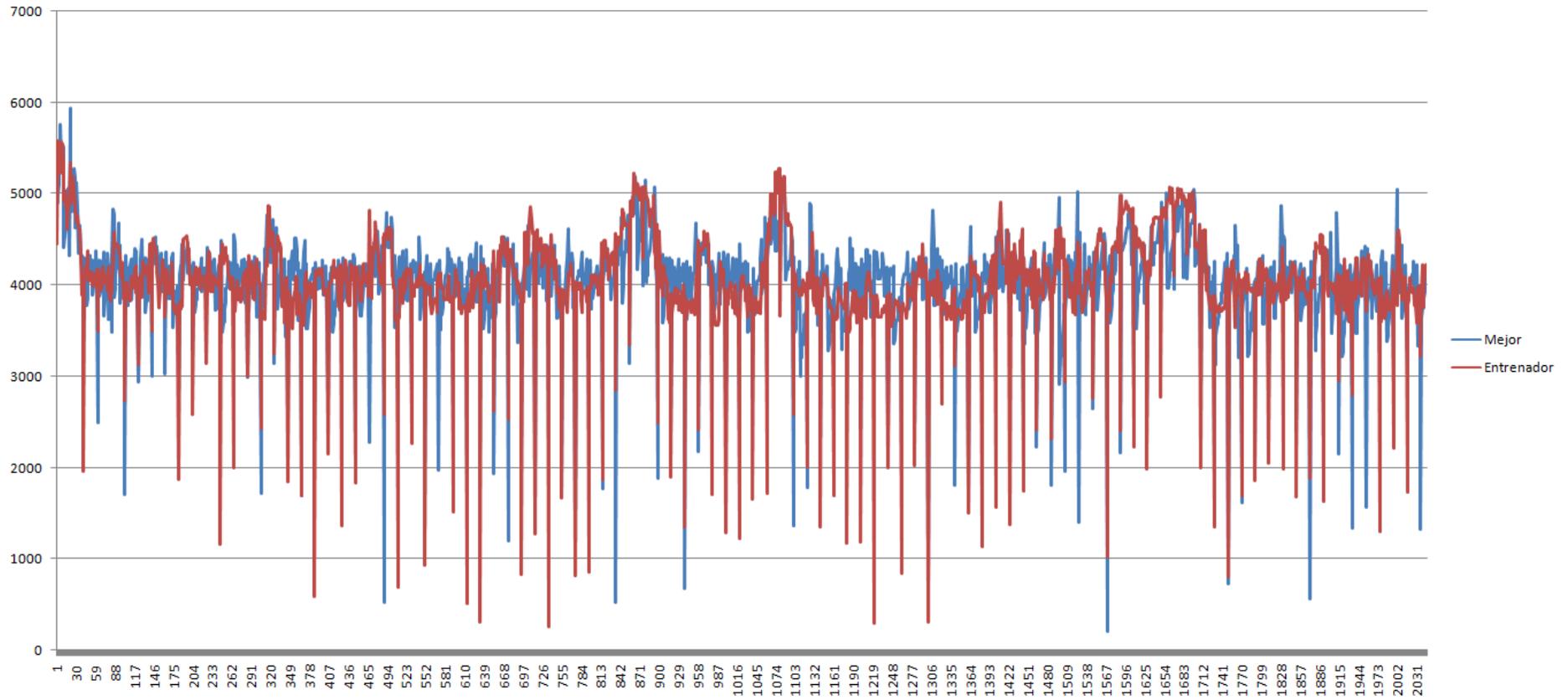


- Grafica de ganancias en la última generación.



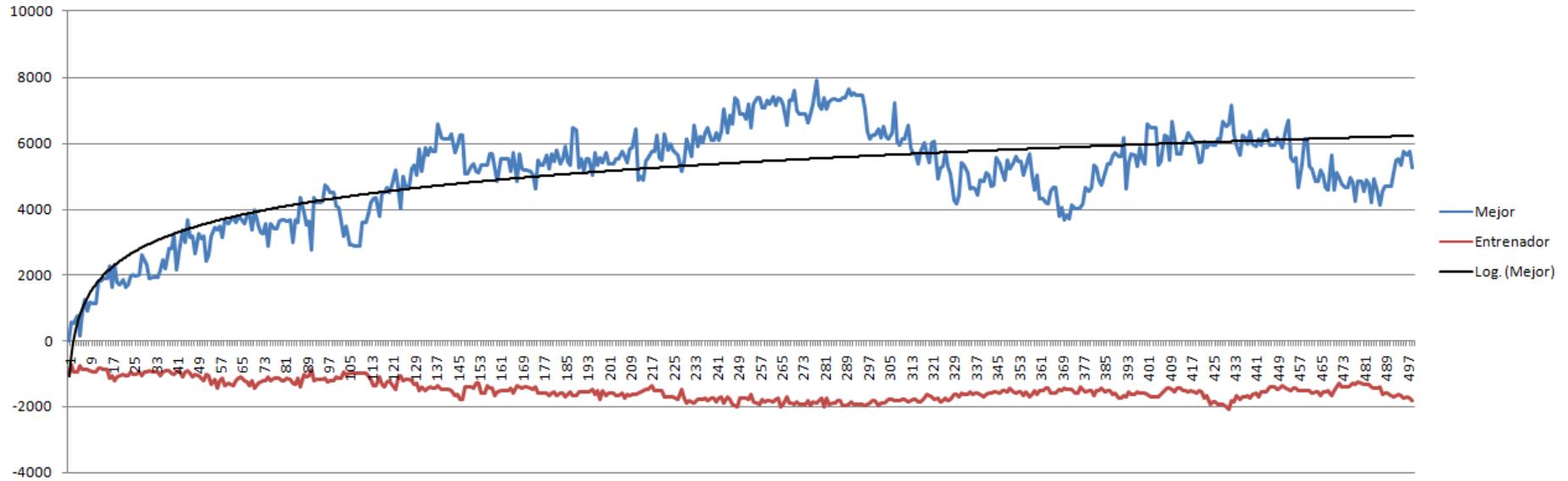


- Grafica de ganancias durante la ejecución.



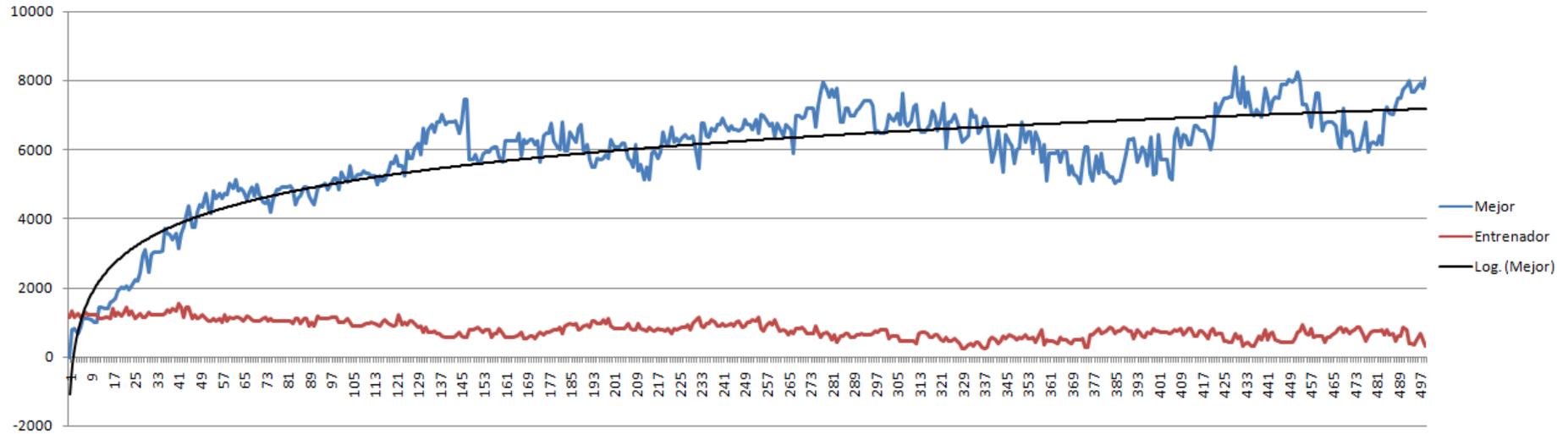


- Grafica de evolución frente a entrenadores aleatorios



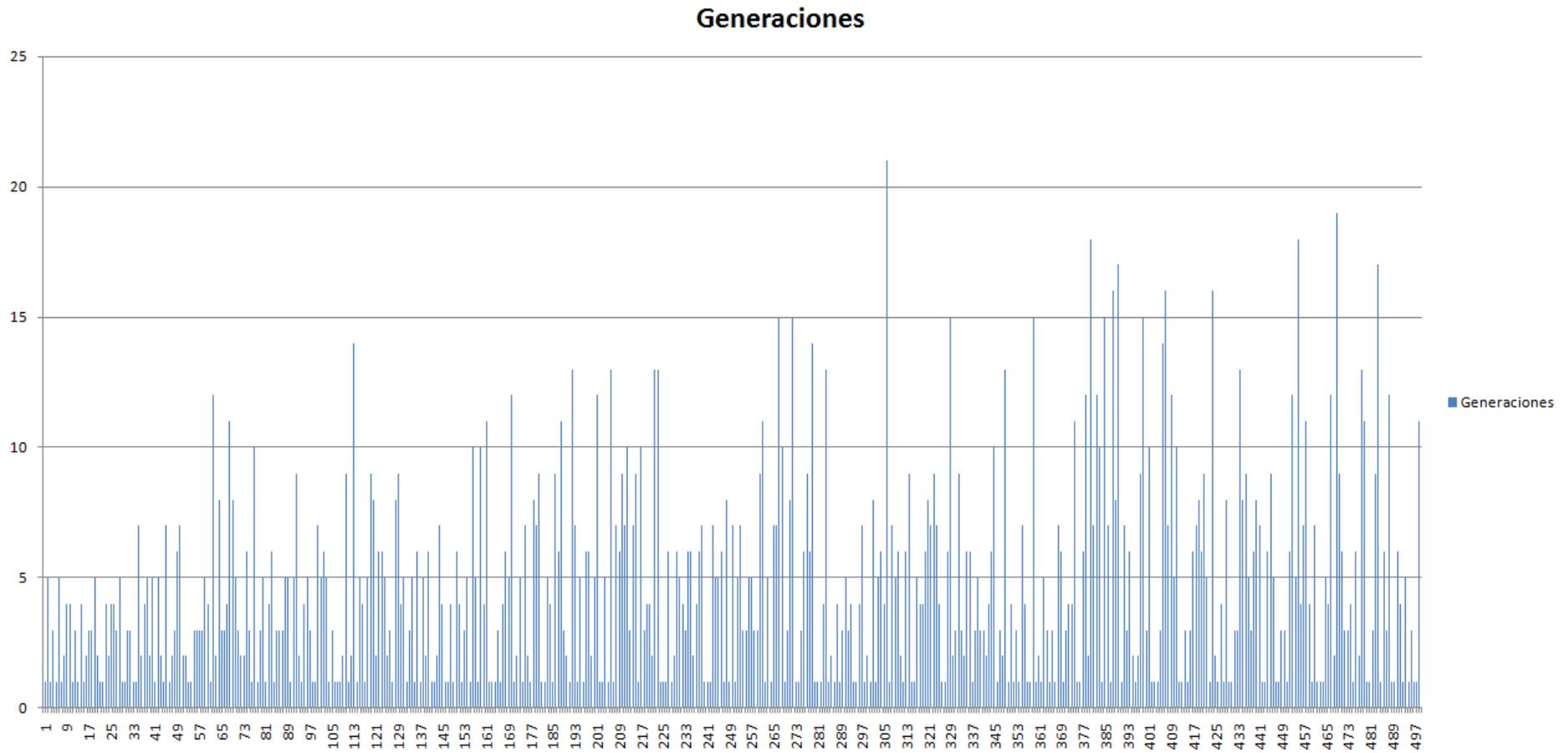


- Grafica de evolución frente a entrenadores pre generados.





- Grafica de coste generacional por entrenador.





5.4.4 Entrenador 2

- 20 mejores manos.

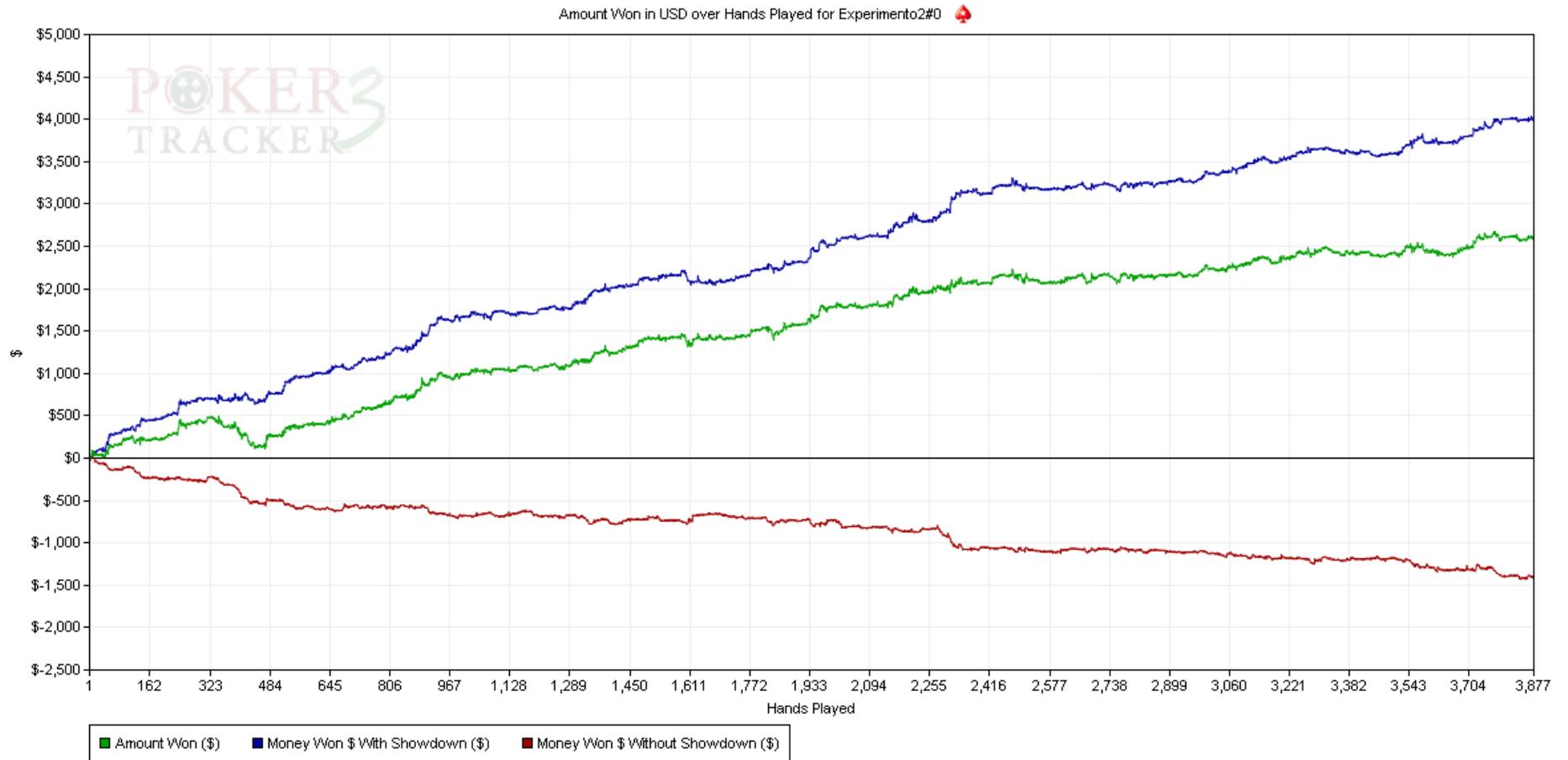
Hand	Times	Win %	Amount Won
KQo	230	50.43	\$1,140.98
AA	56	87.50	\$706.50
AQo	160	49.38	\$533.75
QTo	99	49.49	\$460.75
99	50	42.00	\$449.25
JTo	122	44.26	\$403.75
88	85	51.76	\$354.75
A9o	60	56.67	\$338.50
JJ	62	66.13	\$326.75
KTs	60	55.00	\$280.50
AJs	44	52.27	\$278.50
KK	27	81.48	\$271.00
K9s	22	59.09	\$250.00
AKs	30	93.33	\$249.23
QQ	52	51.92	\$182.25
KQs	24	66.67	\$156.50
Q9o	18	38.89	\$137.25
86s	8	50.00	\$134.75
ATs	35	54.29	\$123.75
A8o	39	58.97	\$103.75

- Combinaciones.

Final Hand	Times	Win %	Amount Won
Four of a Kind	11	90.91	\$331.00
Full House	122	86.07	\$2,136.25
Flush	58	84.48	\$591.00
Straight	187	76.47	\$2,811.50
Three of a Kind	177	61.02	\$920.72
Two Pair	930	47.96	\$1,638.81
One Pair	1,593	38.48	-\$2,815.44
High Card	799	17.65	-\$3,003.54

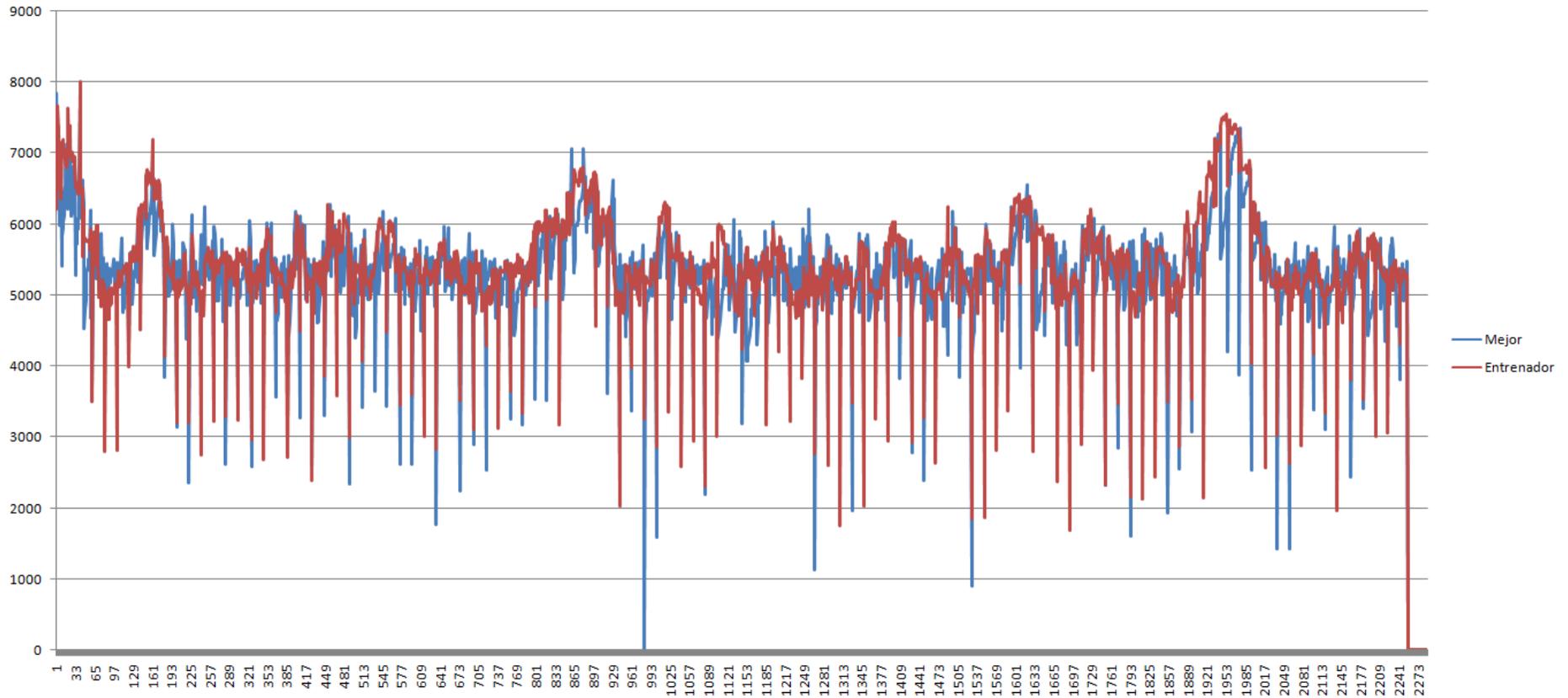


- Grafica de ganancias en la última generación.



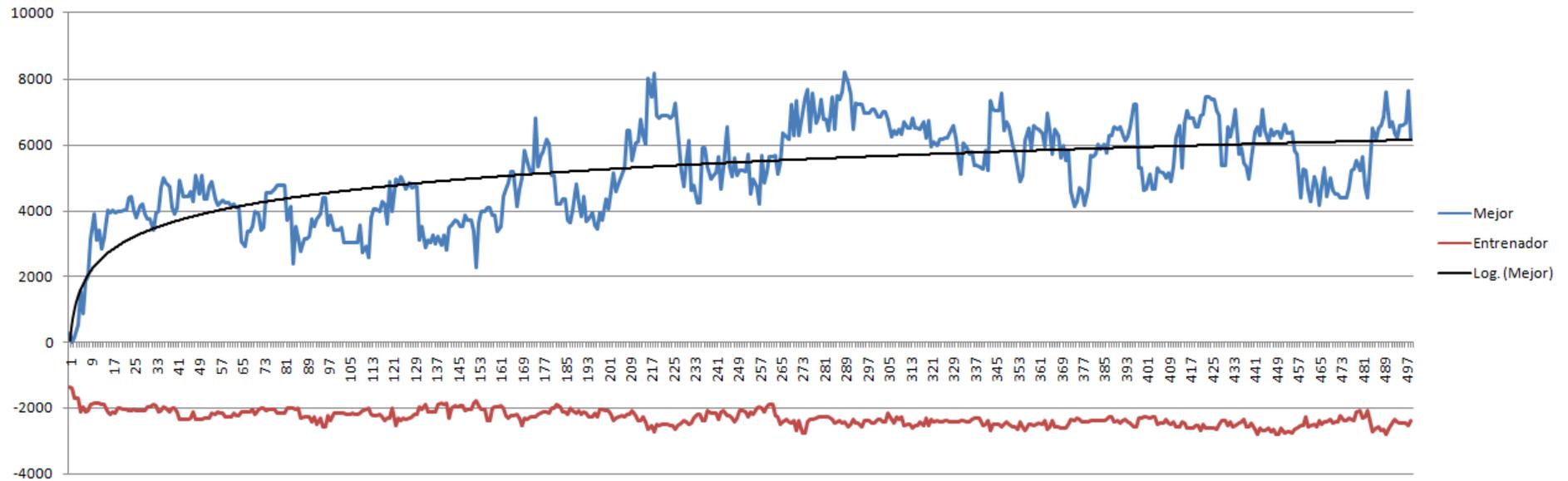


- Grafica de ganancias durante la ejecución.



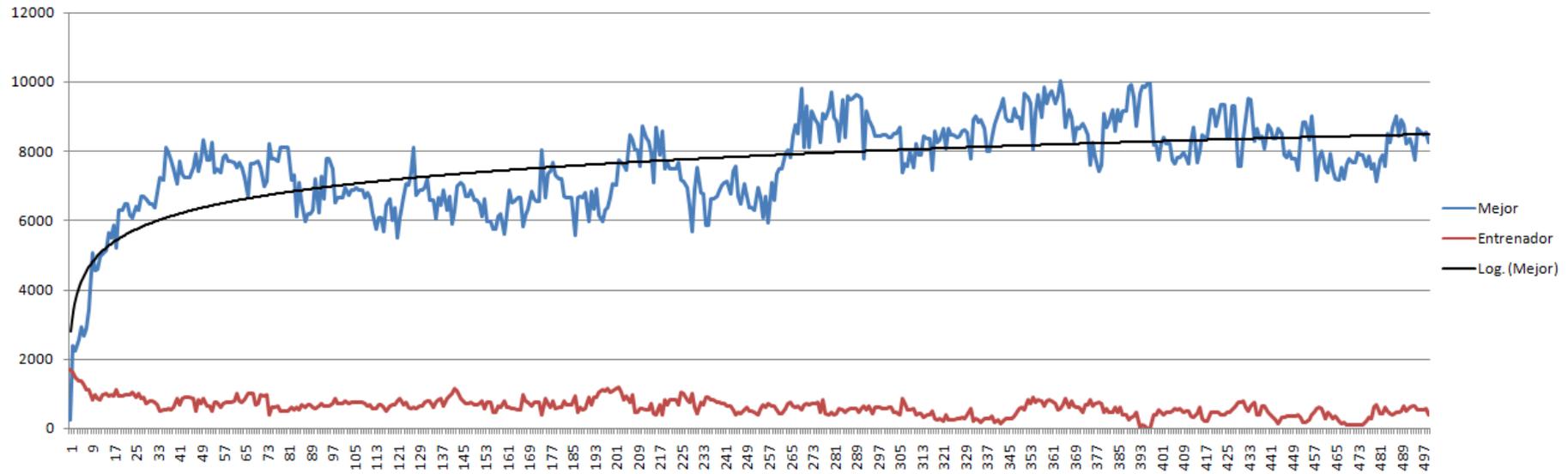


- Grafica de evolución frente a entrenadores aleatorios



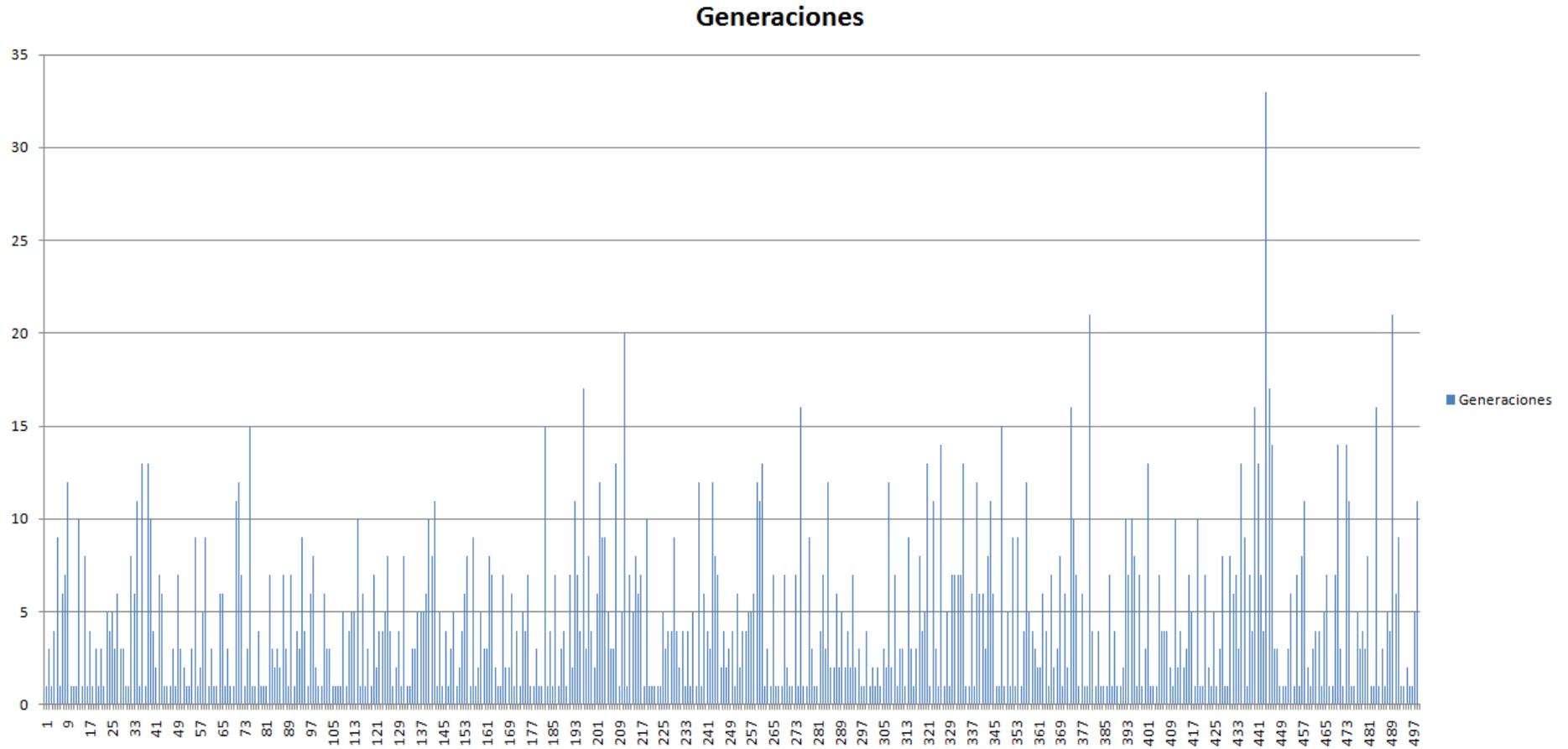


- Grafica de evolución frente a entrenadores pre generados.





- Grafica de coste generacional por entrenador.





5.4.5 Entrenador 3

- 20 mejores manos.

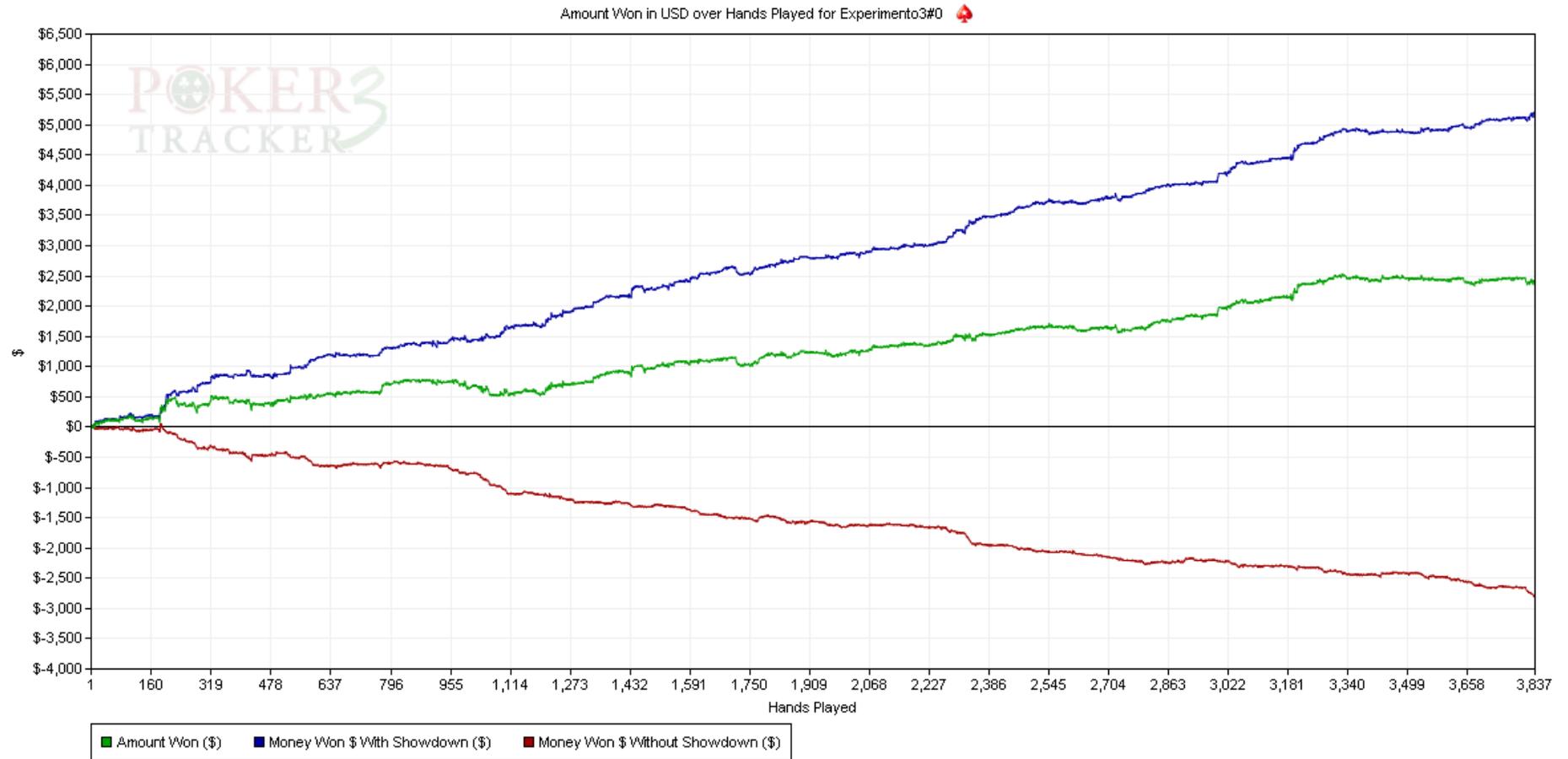
Hand	Times	Win %	Amount Won
AQo	176	46.02	\$935.50
ATo	168	47.62	\$723.25
88	105	52.38	\$707.25
ATs	85	64.71	\$701.99
77	51	68.63	\$487.00
QTo	96	44.79	\$401.00
TT	68	60.29	\$396.75
KQs	87	60.92	\$374.62
QTs	40	80.00	\$370.87
AKo	135	51.85	\$350.75
KK	48	68.75	\$331.75
QJo	75	50.67	\$305.25
99	80	55.00	\$234.75
AA	70	57.14	\$185.75
KTo	82	42.68	\$138.00
33	63	33.33	\$117.50
Q9s	14	50.00	\$97.75
AJs	61	36.07	\$66.50
J8s	6	83.33	\$59.25
QJs	42	45.24	\$57.75

- Combinaciones.

Final Hand	Times	Win %	Amount Won
Straight Flush	1	100.00	\$9.25
Four of a Kind	25	92.00	\$558.75
Full House	169	80.47	\$2,839.75
Flush	114	76.32	\$1,016.74
Straight	161	77.64	\$2,215.00
Three of a Kind	194	70.10	\$2,260.49
Two Pair	872	41.51	-\$499.39
One Pair	1,646	33.66	-\$3,339.17
High Card	655	17.71	-\$2,658.88

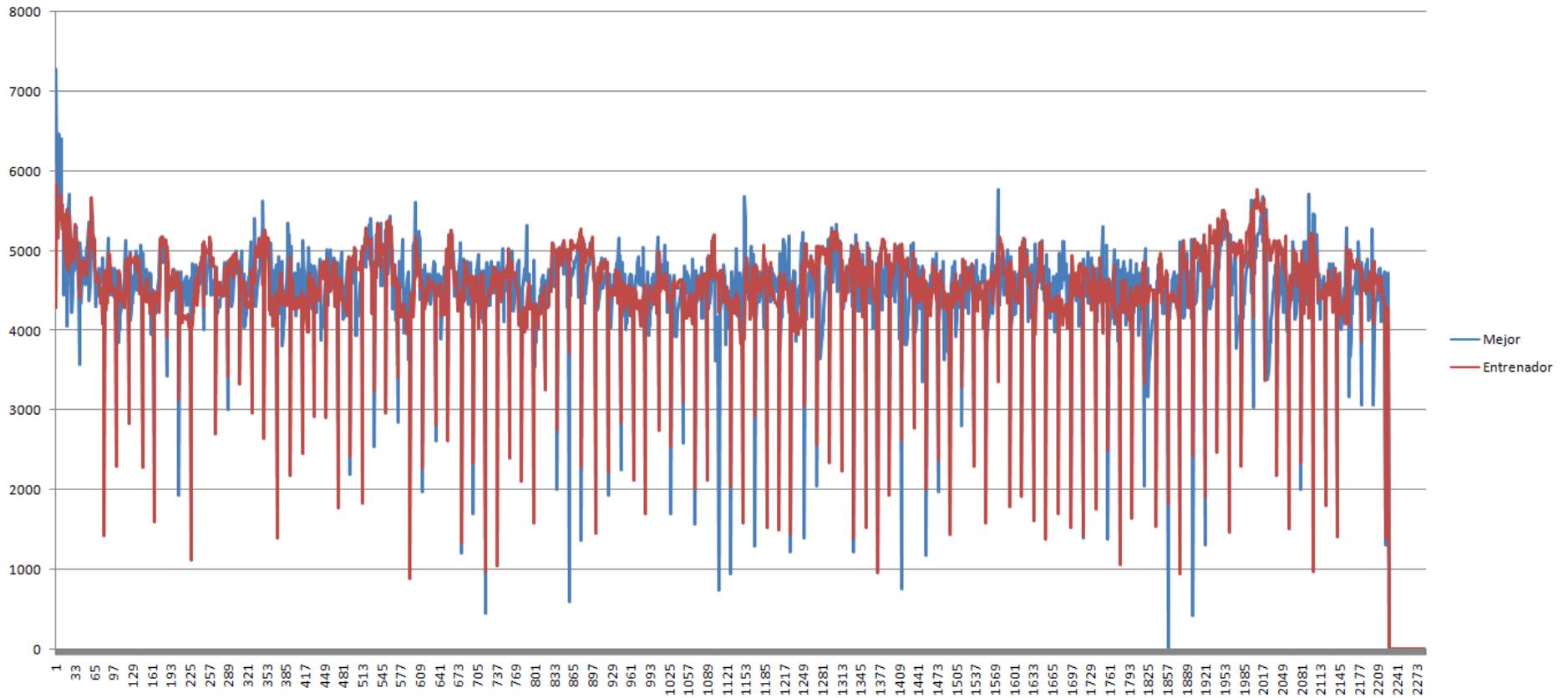


- Grafica de ganancias en la última generación.



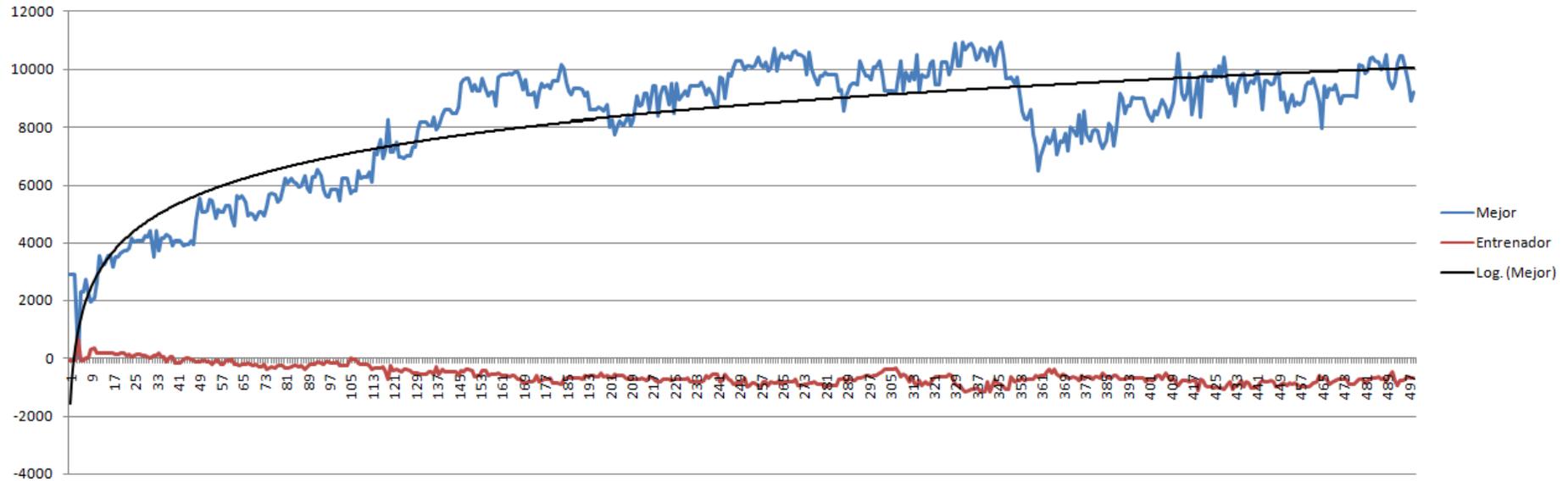


- Grafica de ganancias durante la ejecución.



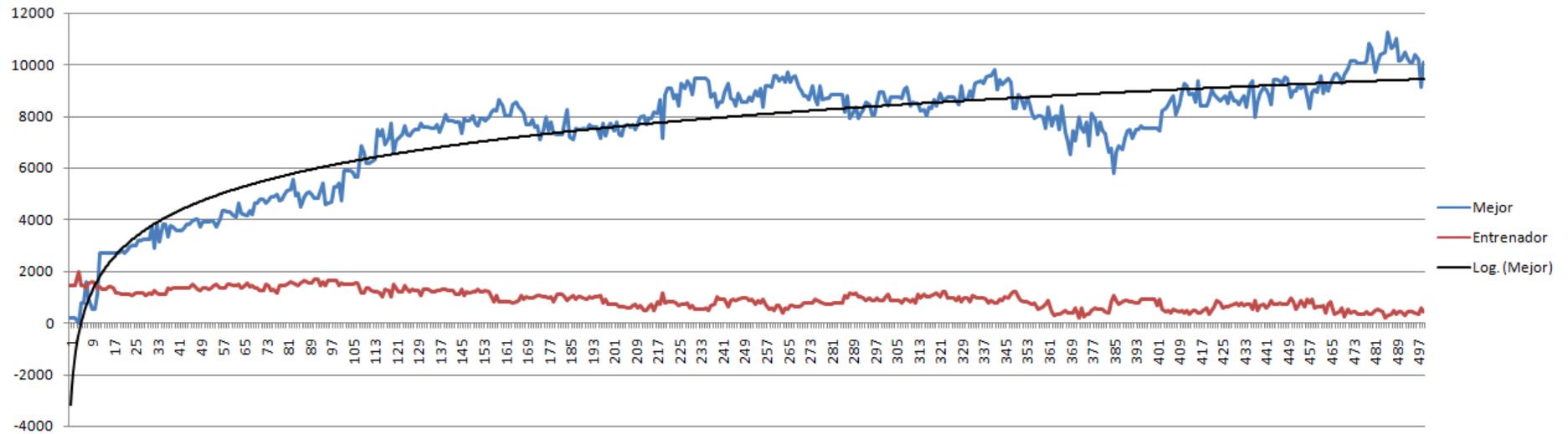


- Grafica de evolución frente a entrenadores aleatorios





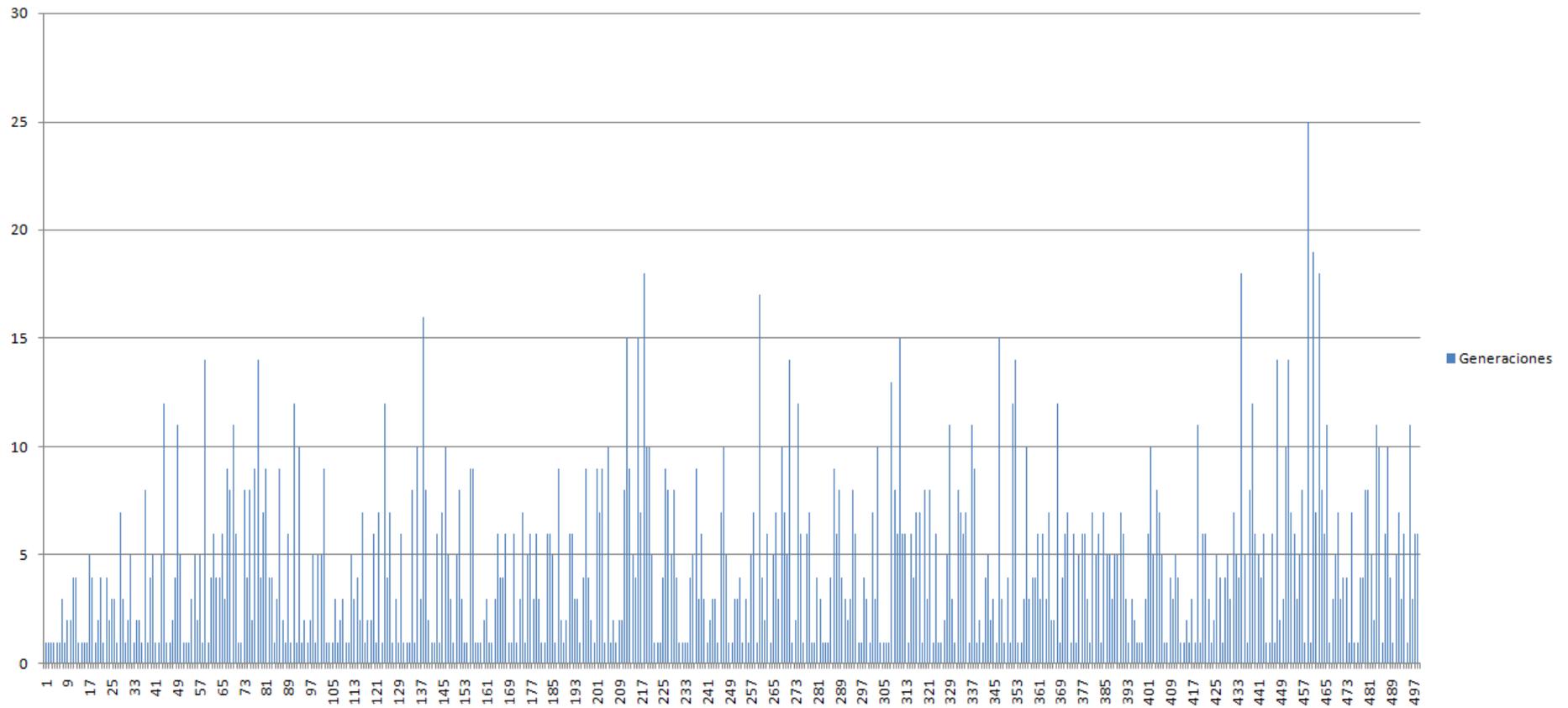
- Grafica de evolución frente a entrenadores pre generados.





- Grafica de coste generacional por entrenador.

Generaciones





5.4.6 Entrenador 4

- 20 mejores manos.

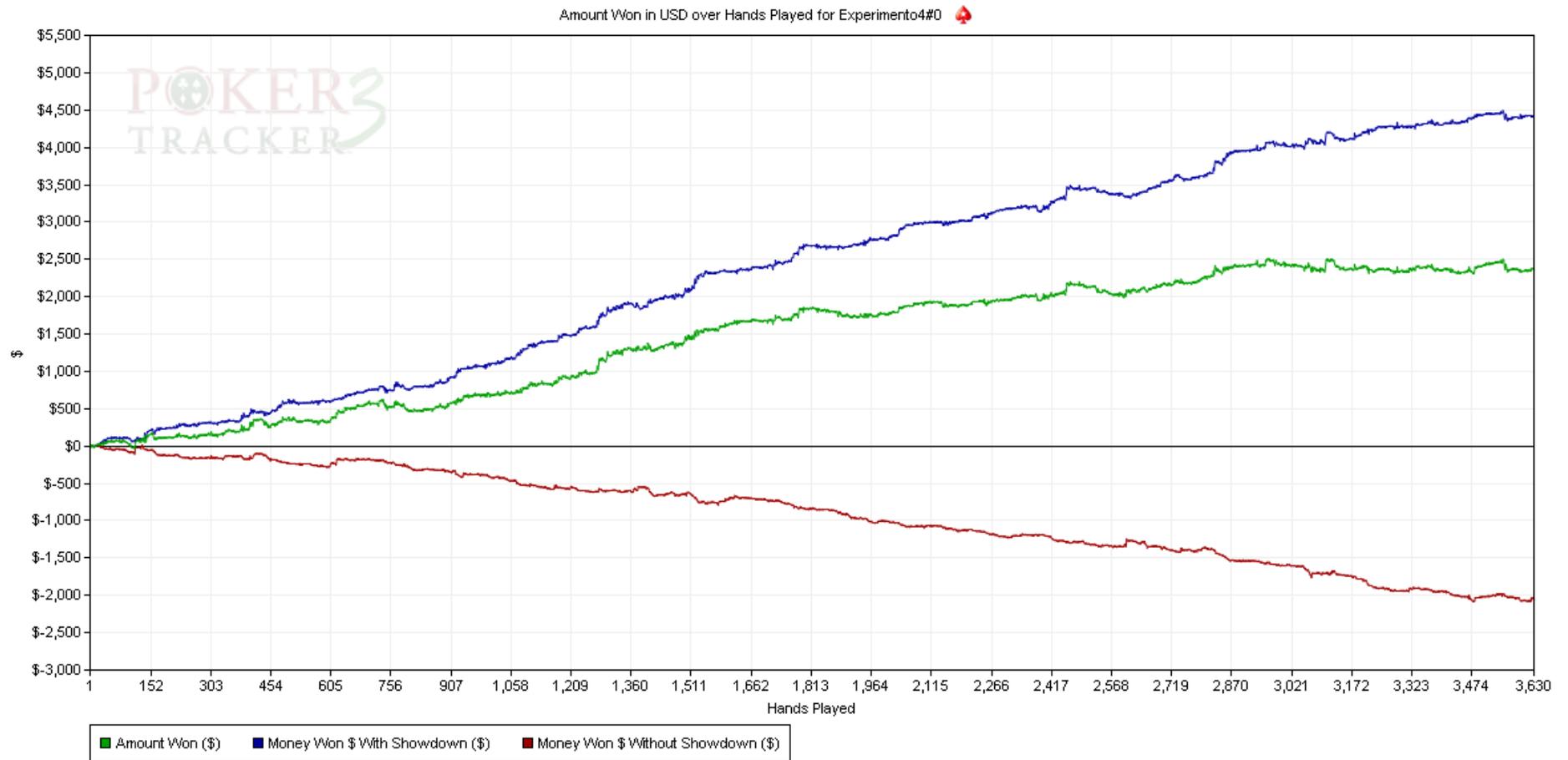
Hand	Times	Win %	Amount Won
KQo	132	56.82	\$1,087.75
AQo	206	45.63	\$565.75
AA	45	88.89	\$498.75
77	63	61.90	\$473.25
AKs	70	61.43	\$405.50
QJo	95	49.47	\$354.62
98o	52	50.00	\$312.62
QQ	79	62.03	\$252.25
QTo	122	46.72	\$251.50
AKo	88	51.14	\$236.50
KJo	111	37.84	\$218.75
44	107	48.60	\$213.00
KK	61	59.02	\$189.50
T9o	41	36.59	\$145.50
A7s	13	61.54	\$109.75
TT	78	47.44	\$95.00
AQs	36	55.56	\$94.75
A5s	26	53.85	\$92.50
99	93	46.24	\$89.00
A8s	17	70.59	\$82.66

- Combinaciones.

Final Hand	Times	Win %	Amount Won
Full House	89	77.53	\$1,357.62
Flush	62	62.90	\$369.75
Straight	175	78.29	\$2,040.19
Three of a Kind	265	67.17	\$1,998.50
Two Pair	1,019	49.66	\$2,541.10
One Pair	1,458	33.74	-\$3,499.02
High Card	562	15.12	-\$2,444.51

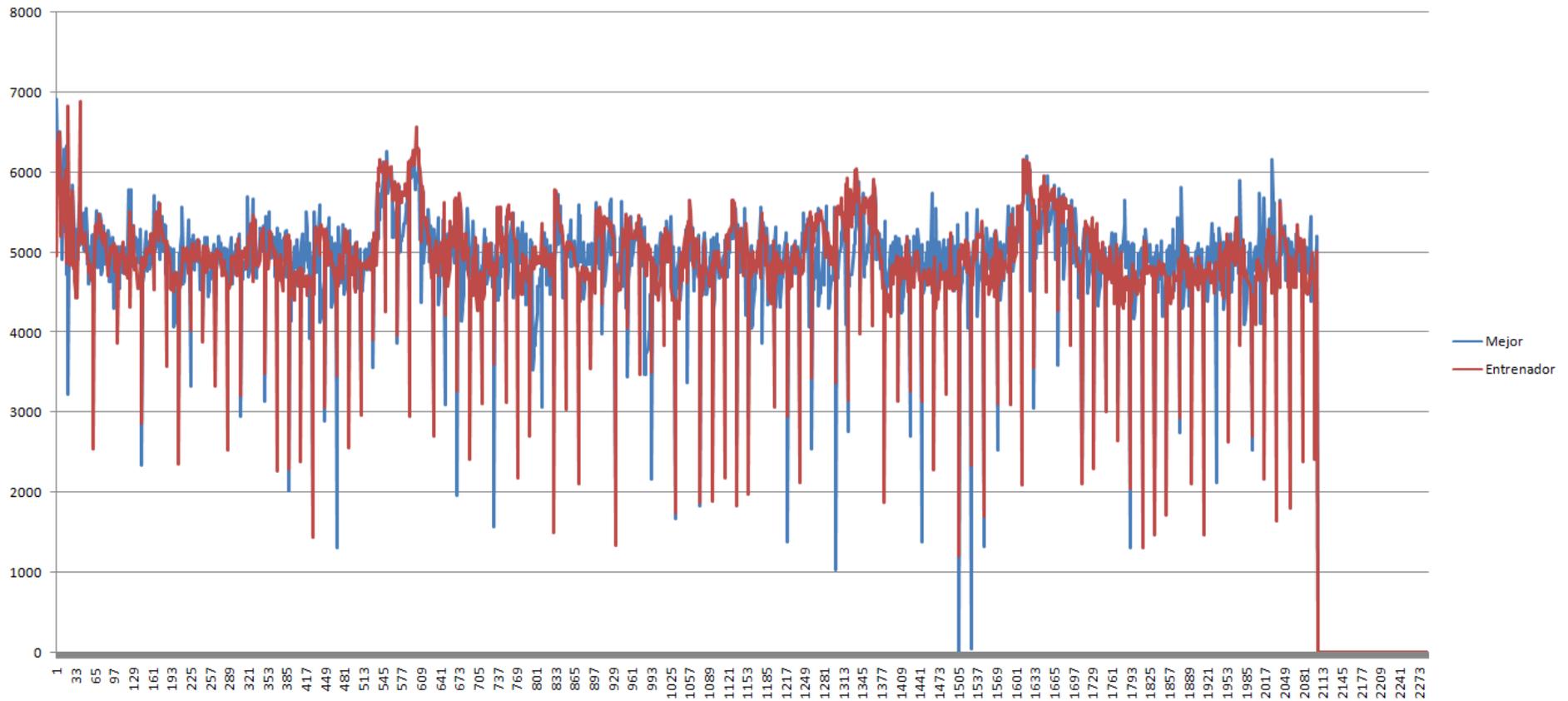


- Grafica de ganancias en la última generación.



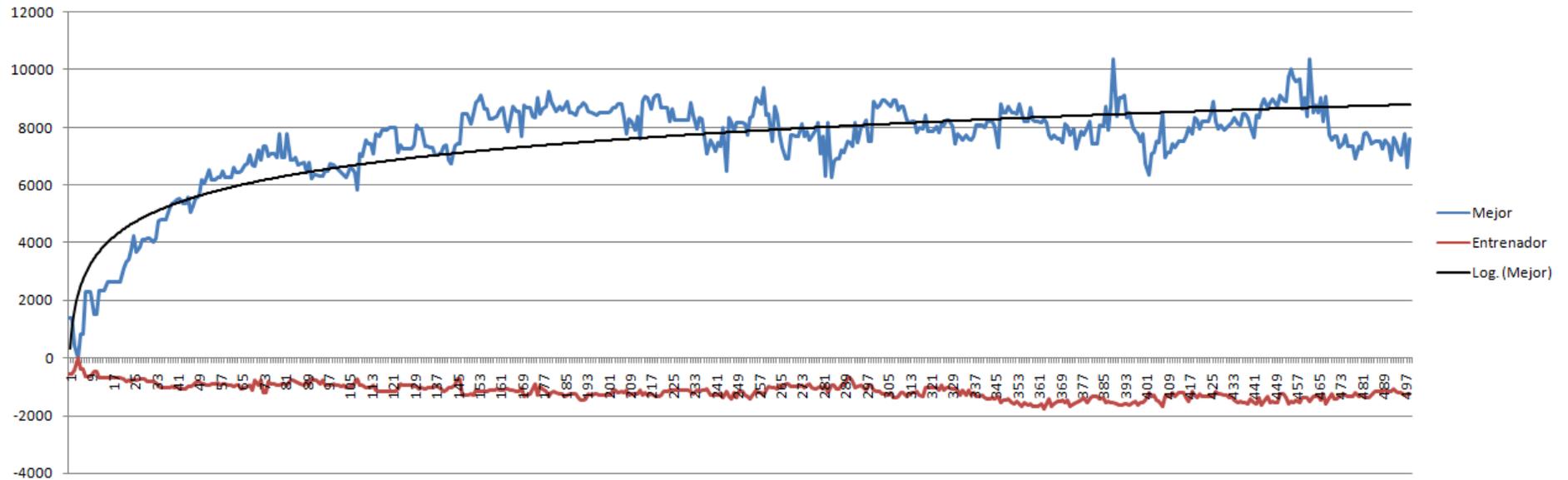


- Grafica de ganancias durante la ejecución.



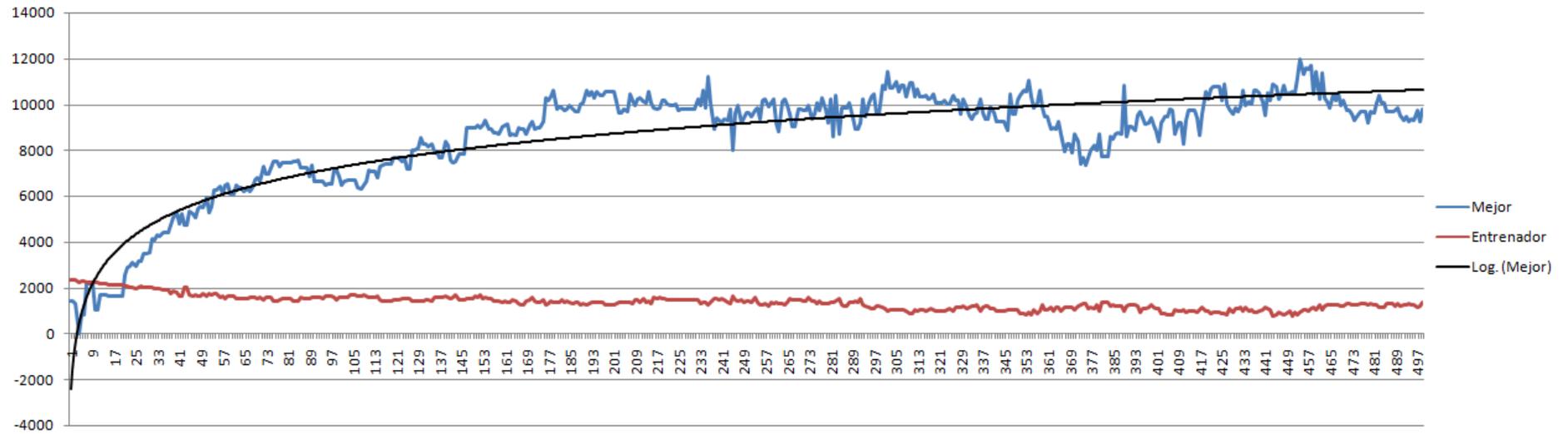


- Grafica de evolución frente a entrenadores aleatorios



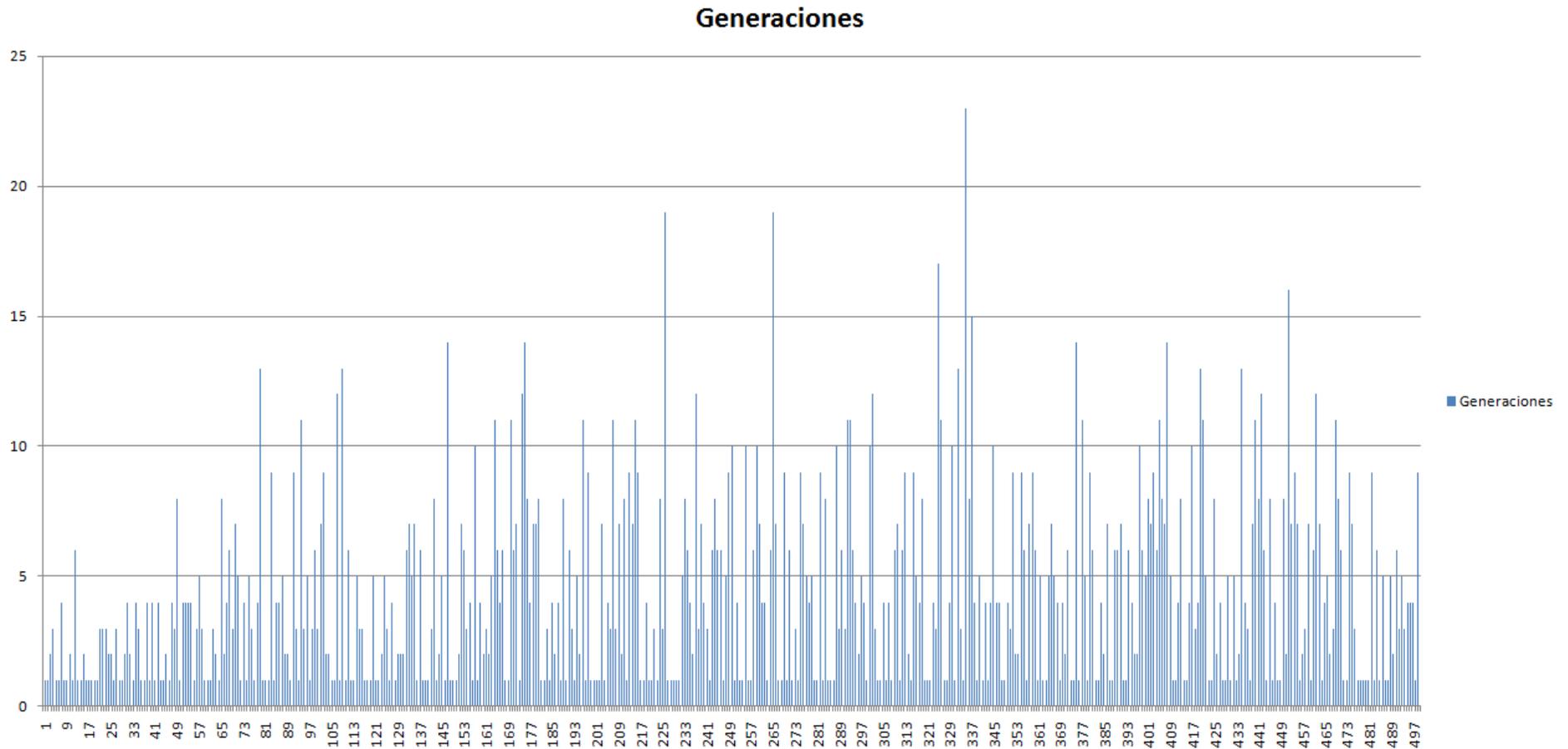


- Grafica de evolución frente a entrenadores pre generados.





- Grafica de coste generacional por entrenador.





5.4.7 Entrenador 5

- 20 mejores manos.

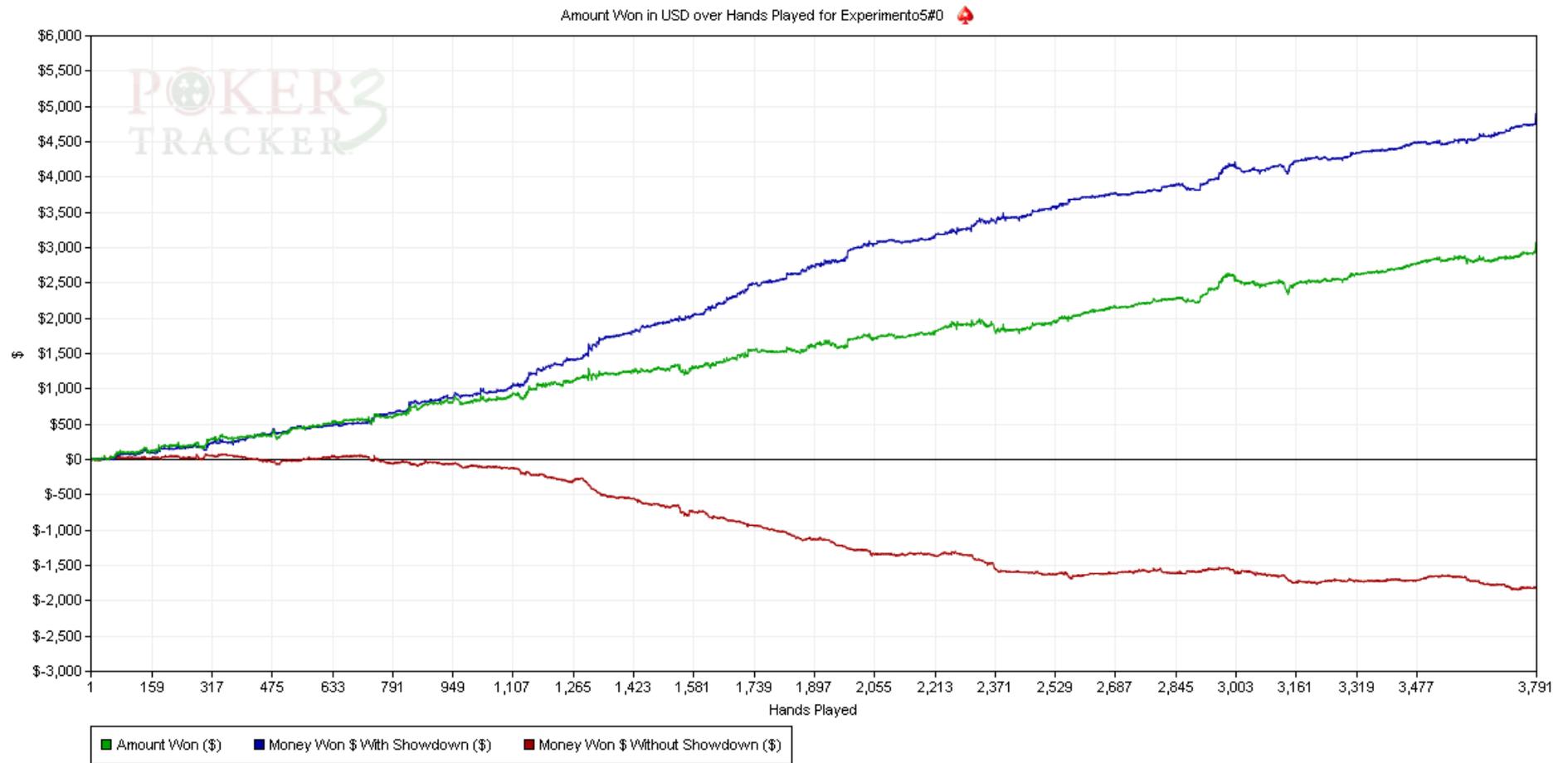
Hand	Times	Win %	Amount Won
AQo	172	66.28	\$1,067.25
AA	47	91.49	\$742.36
66	103	64.08	\$703.50
JJ	67	58.21	\$592.25
AKo	122	61.48	\$558.24
ATo	190	53.68	\$513.54
AJs	94	57.45	\$453.87
KQo	89	55.06	\$350.00
KK	70	64.29	\$311.00
AQs	65	52.31	\$293.75
AKs	63	66.67	\$184.37
JTs	27	55.56	\$170.50
J9o	58	50.00	\$131.00
K9s	8	75.00	\$119.00
KTs	48	31.25	\$108.75
54o	45	35.56	\$108.25
A9s	25	52.00	\$108.00
A8o	38	36.84	\$91.37
QJs	24	45.83	\$81.50
97o	13	53.85	\$77.00

- Combinaciones.

Final Hand	Times	Win %	Amount Won
Straight Flush	3	100.00	\$22.25
Four of a Kind	7	100.00	\$185.00
Full House	86	89.53	\$1,454.75
Flush	105	60.00	\$957.25
Straight	128	73.44	\$1,101.22
Three of a Kind	213	74.65	\$2,730.25
Two Pair	1,008	48.51	\$1,435.37
One Pair	1,508	34.48	-\$2,676.29
High Card	733	24.01	-\$2,169.00

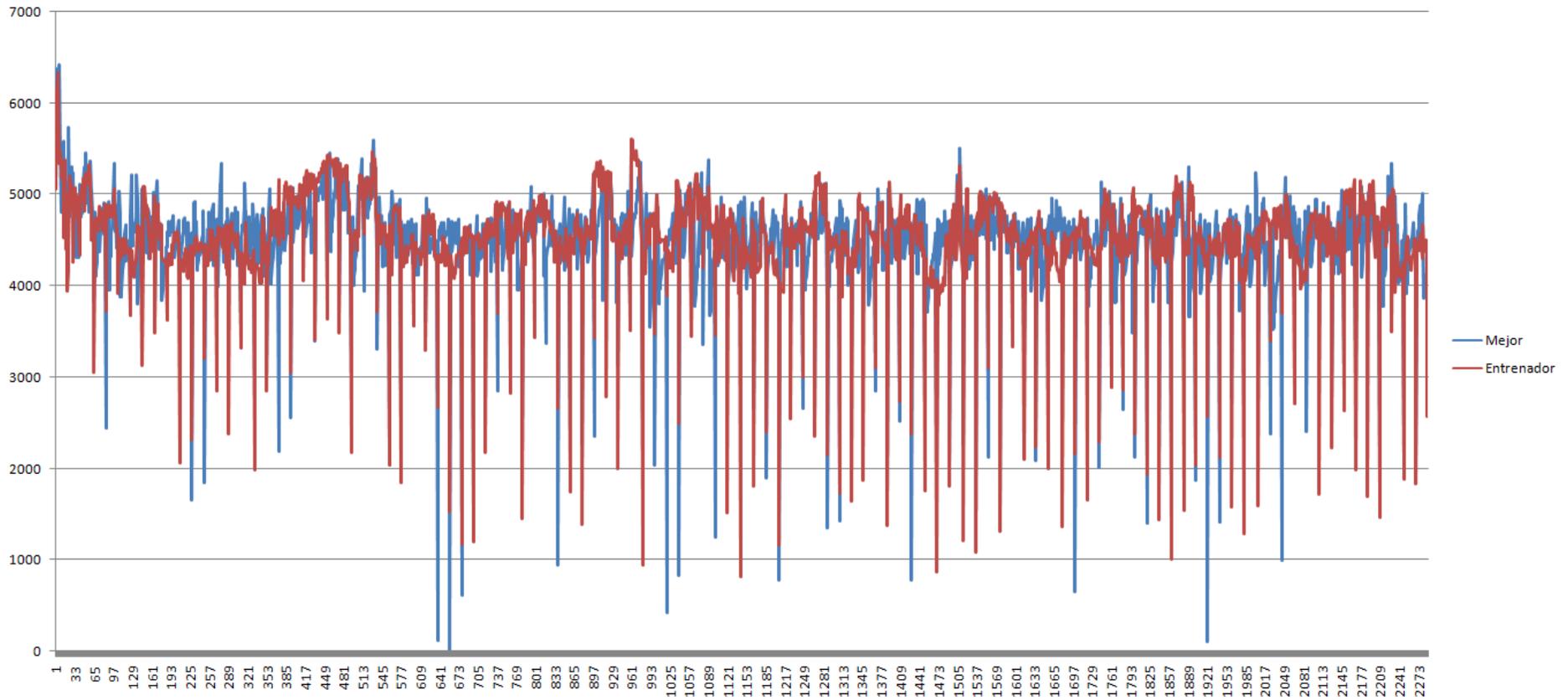


- Grafica de ganancias en la última generación.



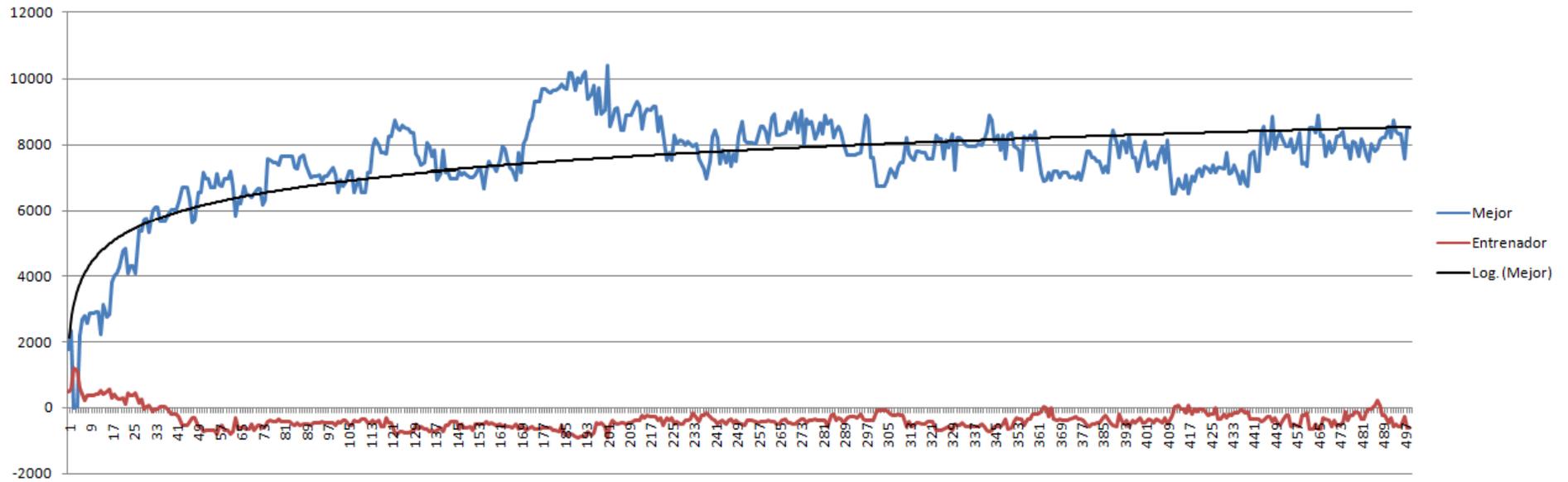


- Grafica de ganancias durante la ejecución.



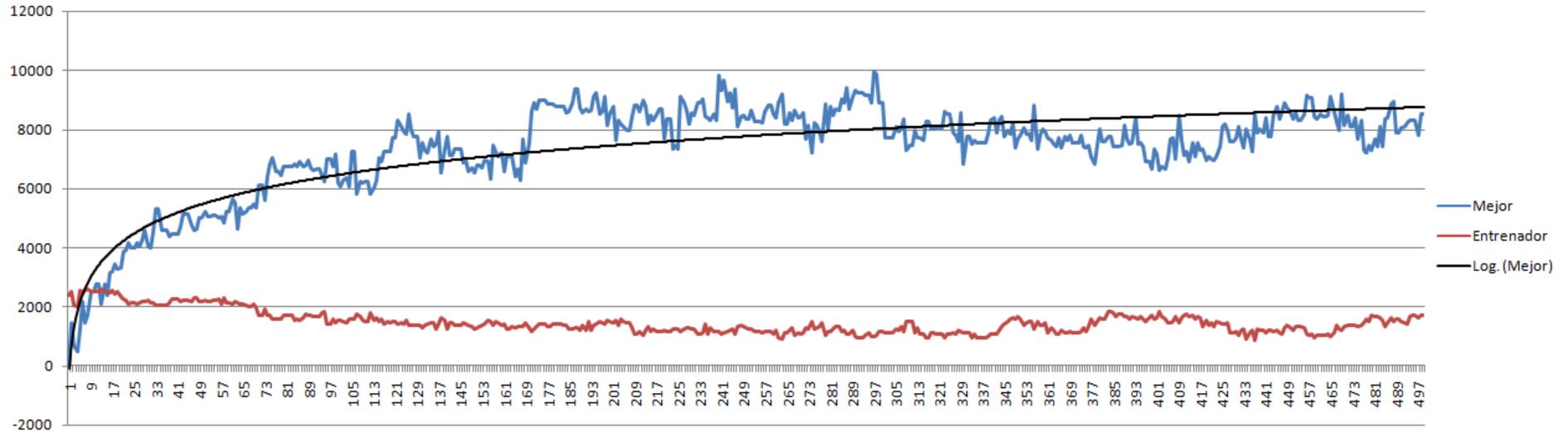


- Grafica de evolución frente a entrenadores aleatorios



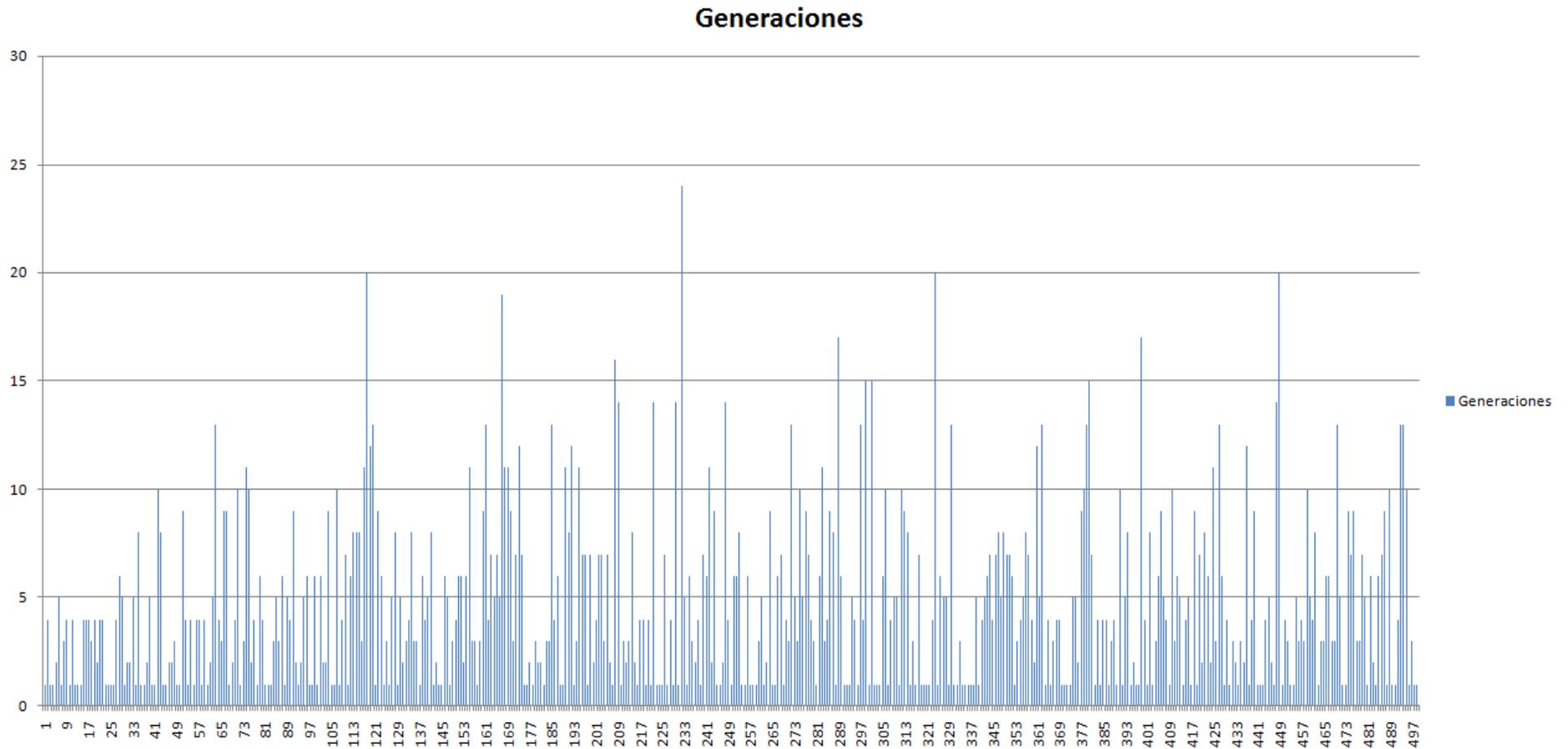


- Grafica de evolución frente a entrenadores pre generados.





- Grafica de coste generacional por entrenador.





5.4.8 Entrenador 6

- 20 mejores manos.

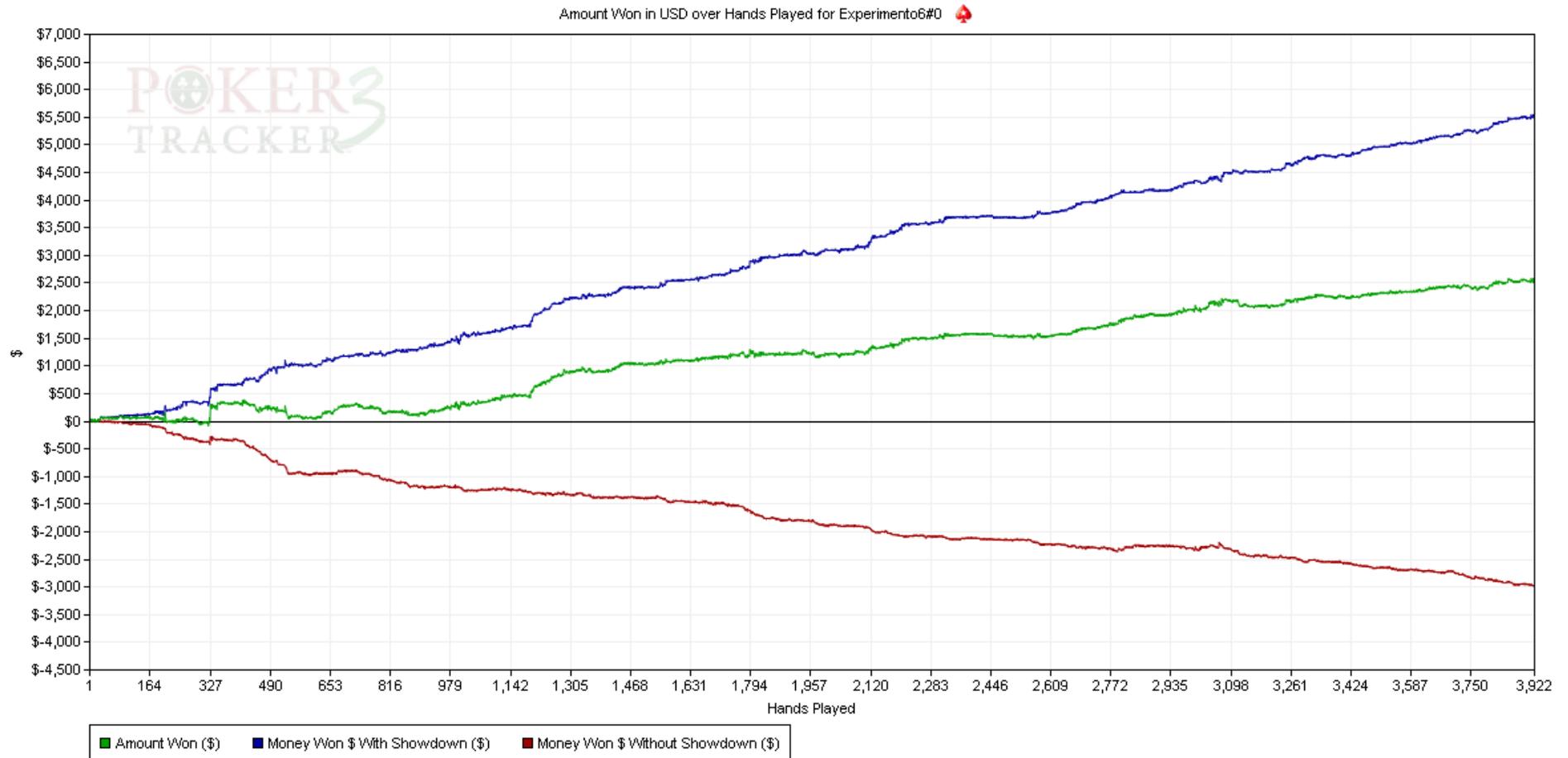
Hand	Times	Win %	Amount Won
KQs	73	80.82	\$1,936.50
AQo	152	60.53	\$1,441.75
AKo	136	60.29	\$815.50
AQs	62	61.29	\$696.75
KK	52	84.62	\$554.25
AJo	237	48.10	\$537.75
55	44	52.27	\$318.00
QJo	82	46.34	\$262.75
QQ	47	61.70	\$240.25
TT	98	45.92	\$172.75
JJ	71	56.34	\$156.75
AA	42	59.52	\$146.50
A5s	20	65.00	\$133.12
A9o	31	61.29	\$129.50
KQo	193	38.86	\$105.50
QJs	40	50.00	\$87.00
88	142	42.25	\$84.25
A7s	22	18.18	\$64.00
76s	16	56.25	\$53.75
75s	11	36.36	\$53.00

- Combinaciones.

Final Hand	Times	Win %	Amount Won
Four of a Kind	14	85.71	\$262.00
Full House	132	71.21	\$1,283.68
Flush	90	81.11	\$1,894.75
Straight	175	76.57	\$1,663.74
Three of a Kind	218	69.27	\$2,445.00
Two Pair	1,065	48.36	\$1,142.74
One Pair	1,518	26.35	-\$4,072.00
High Card	710	26.90	-\$2,040.26

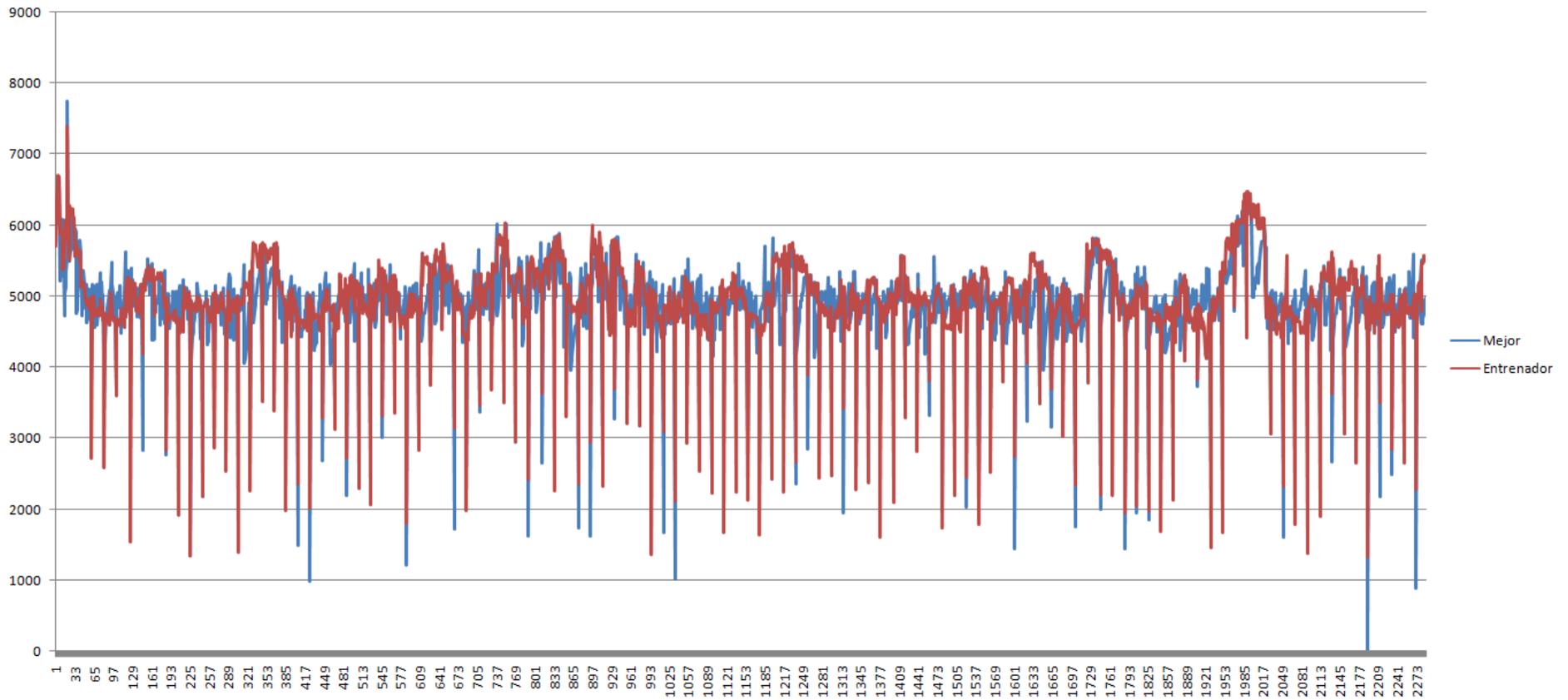


- Grafica de ganancias en la última generación.



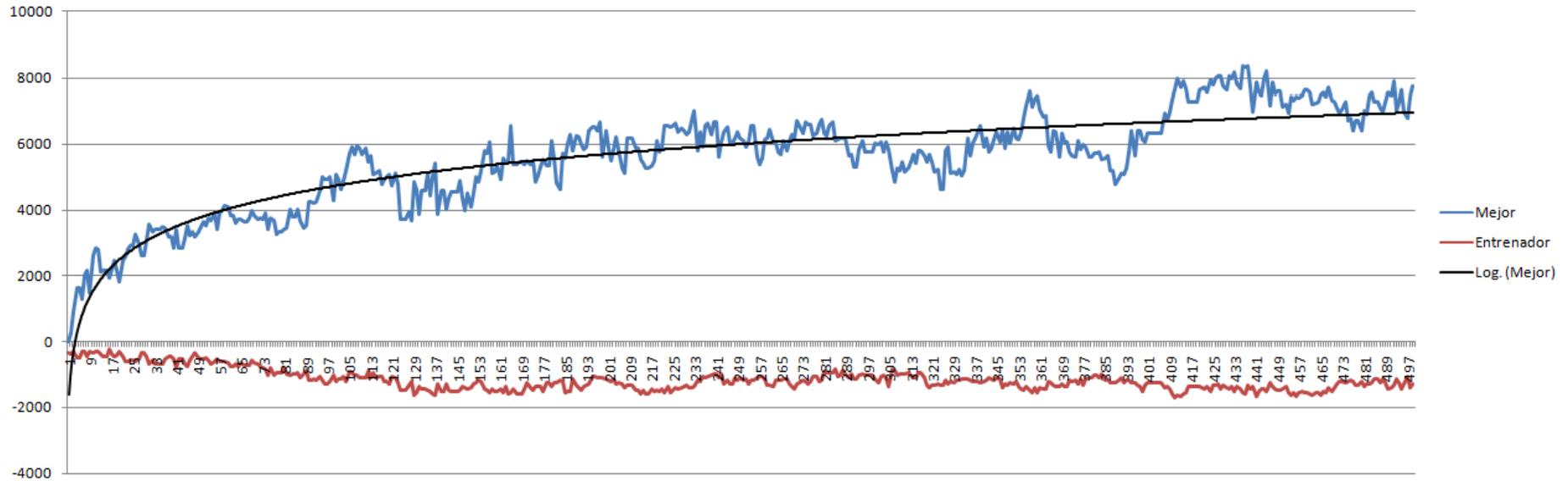


- Grafica de ganancias durante la ejecución.



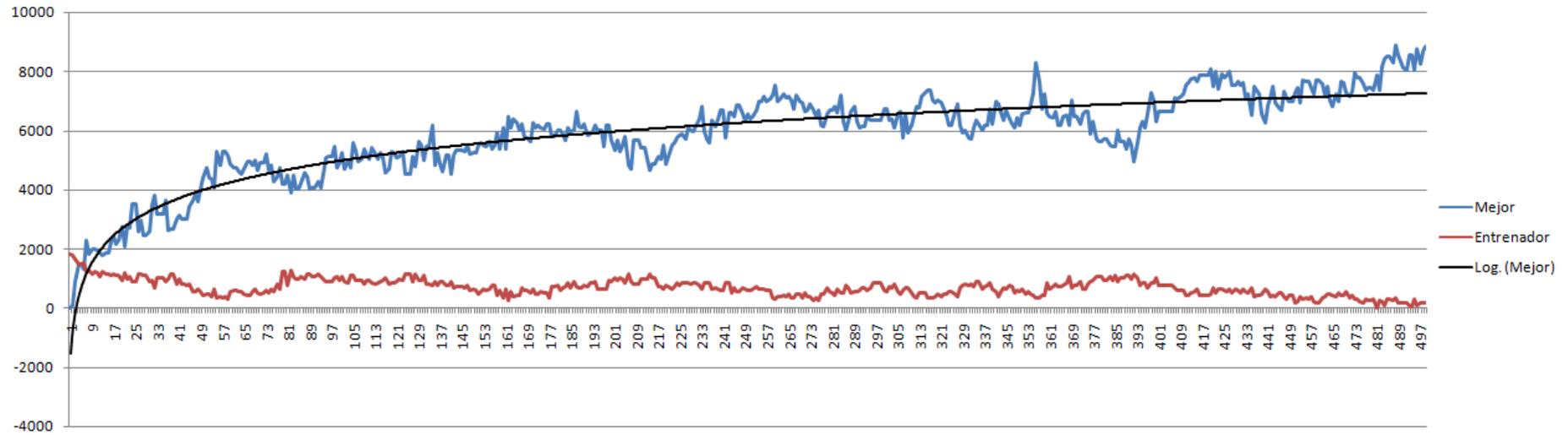


- Grafica de evolución frente a entrenadores aleatorios



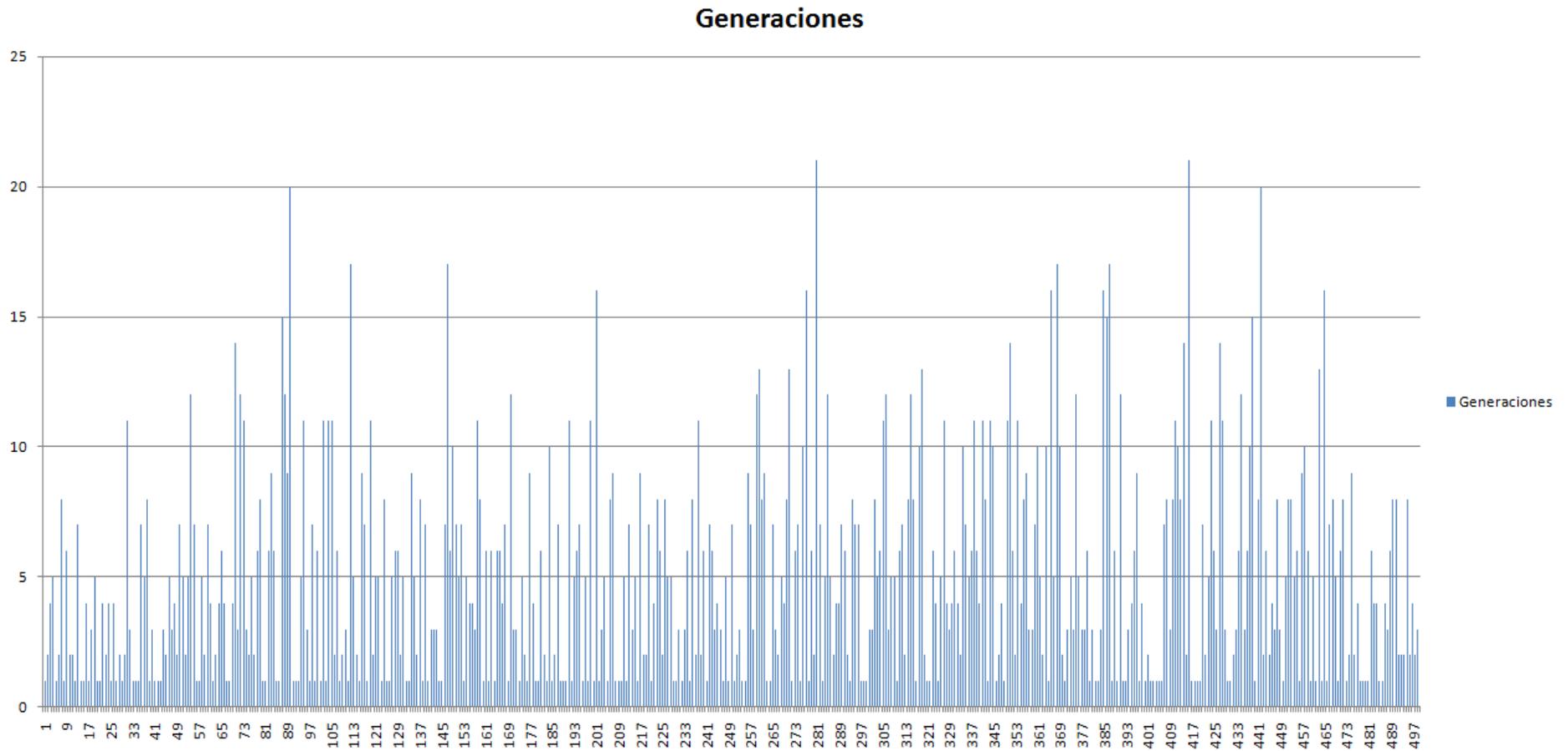


- Grafica de evolución frente a entrenadores pre generados.





- Grafica de coste generacional por entrenador.





5.4.9 Entrenador 7

- 20 mejores manos.

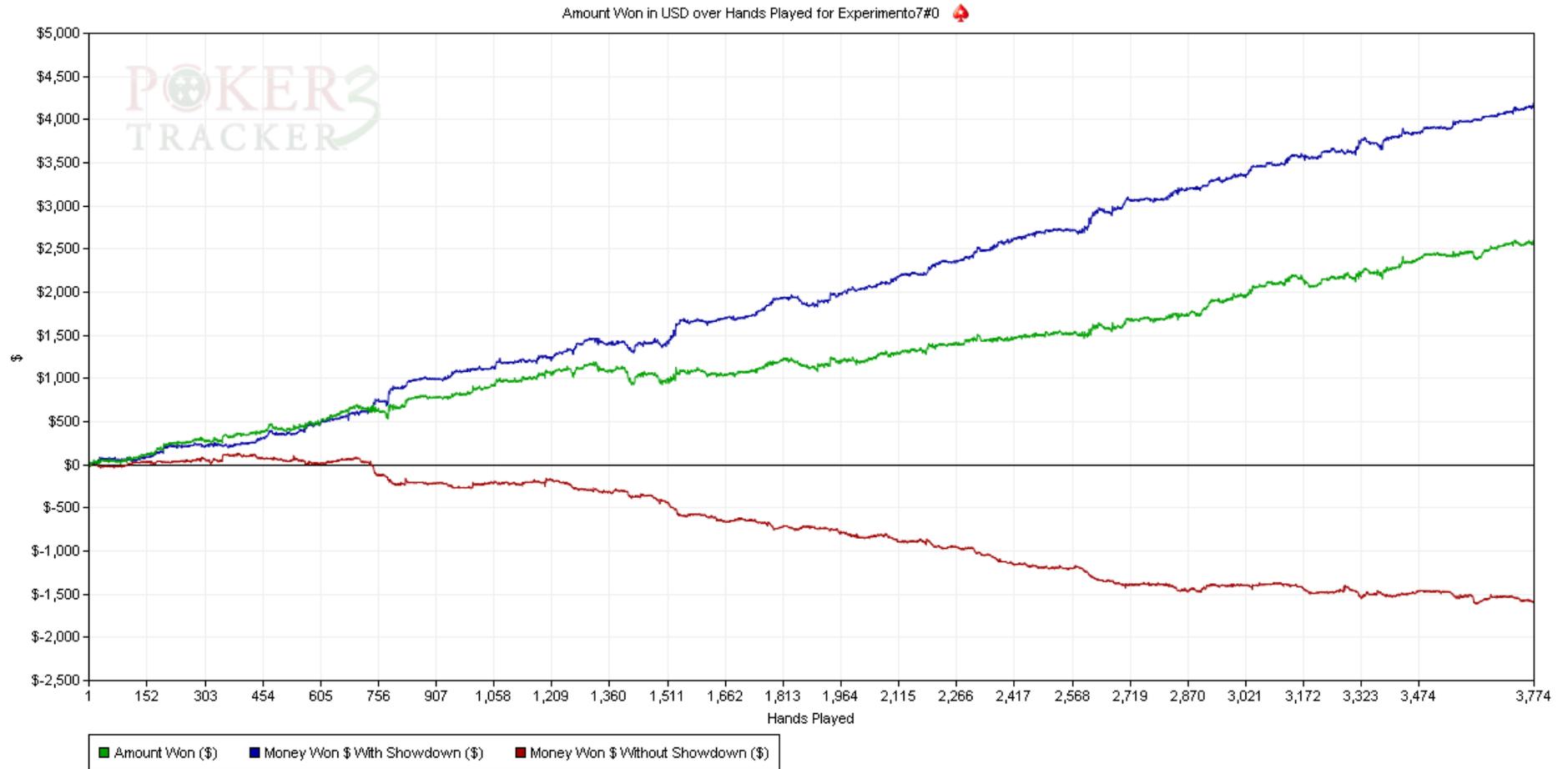
Hand	Times	Win %	Amount Won
AA	66	92.42	\$755.25
AKo	140	62.86	\$549.24
22	64	53.13	\$538.75
44	88	40.91	\$532.25
55	108	52.78	\$518.25
KJo	140	54.29	\$319.33
KK	40	50.00	\$285.50
QJo	80	47.50	\$242.50
ATs	37	56.76	\$145.75
KTs	35	45.71	\$126.25
KQs	63	47.62	\$125.25
66	122	45.08	\$124.50
QTo	93	41.94	\$122.00
J9o	42	52.38	\$114.47
A6s	19	52.63	\$108.00
QQ	36	63.89	\$107.50
33	61	36.07	\$105.50
43o	13	61.54	\$92.50
ATo	153	44.44	\$75.85
QJs	32	53.13	\$55.25
97o	21	38.10	\$51.75

- Combinaciones.

Final Hand	Times	Win %	Amount Won
Straight Flush	4	100.00	\$47.50
Full House	155	81.29	\$2,276.25
Flush	99	61.62	\$690.08
Straight	102	79.41	\$1,235.99
Three of a Kind	277	75.45	\$3,255.50
Two Pair	799	42.80	-\$800.13
One Pair	1,565	36.81	-\$1,332.47
High Card	773	23.67	-\$2,768.02

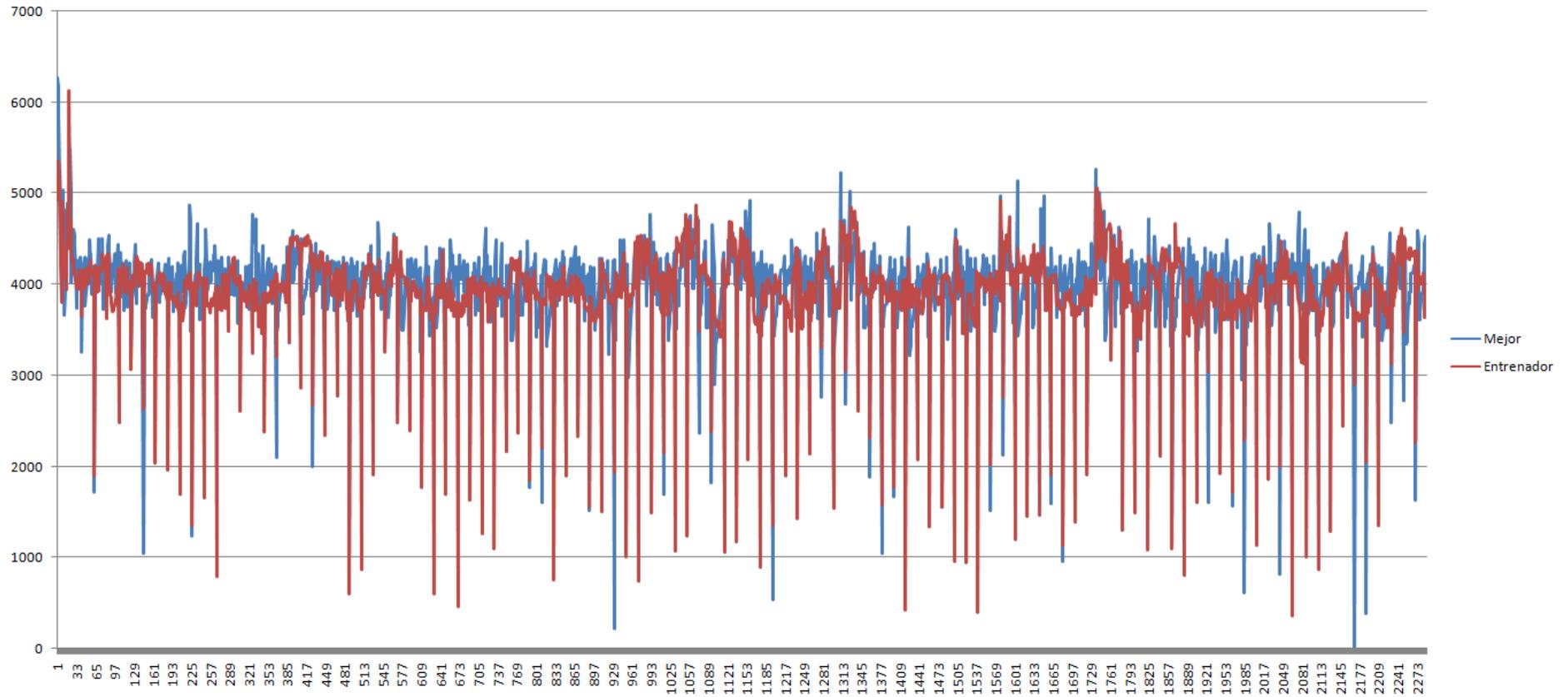


- Grafica de ganancias en la última generación.



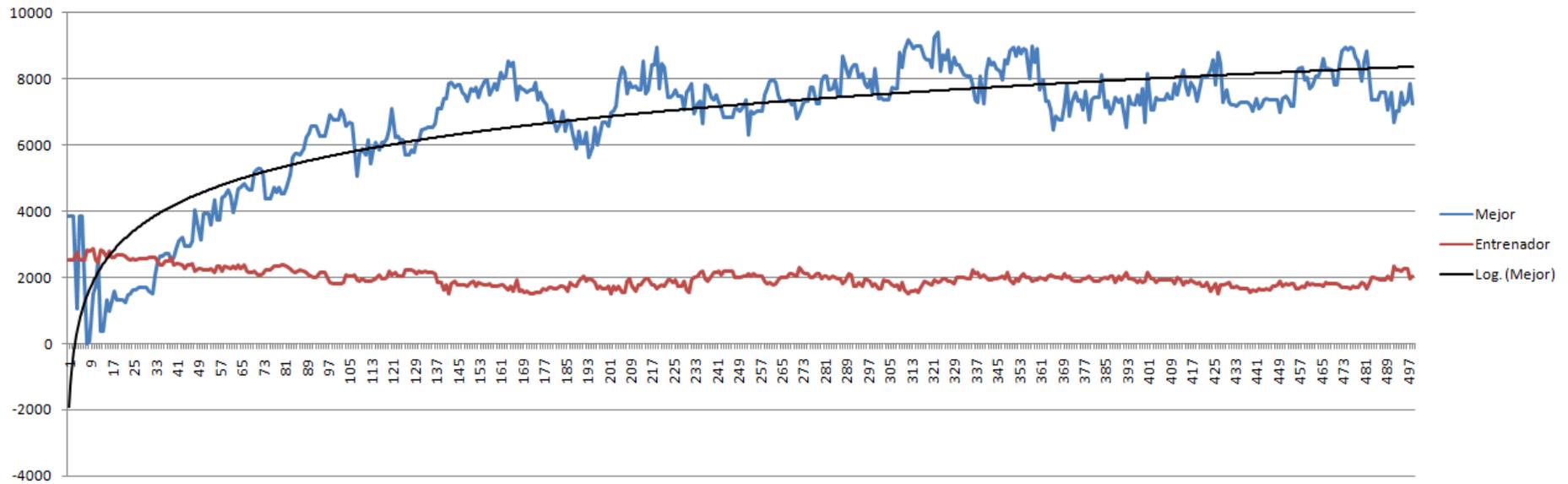


- Grafica de ganancias durante la ejecución.



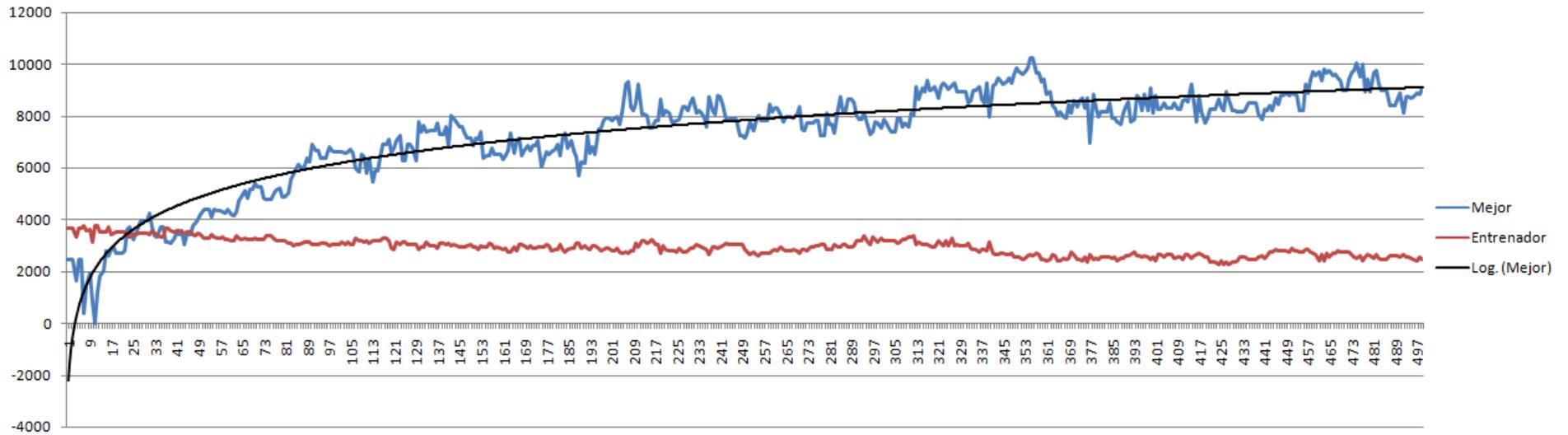


- Grafica de evolución frente a entrenadores aleatorios



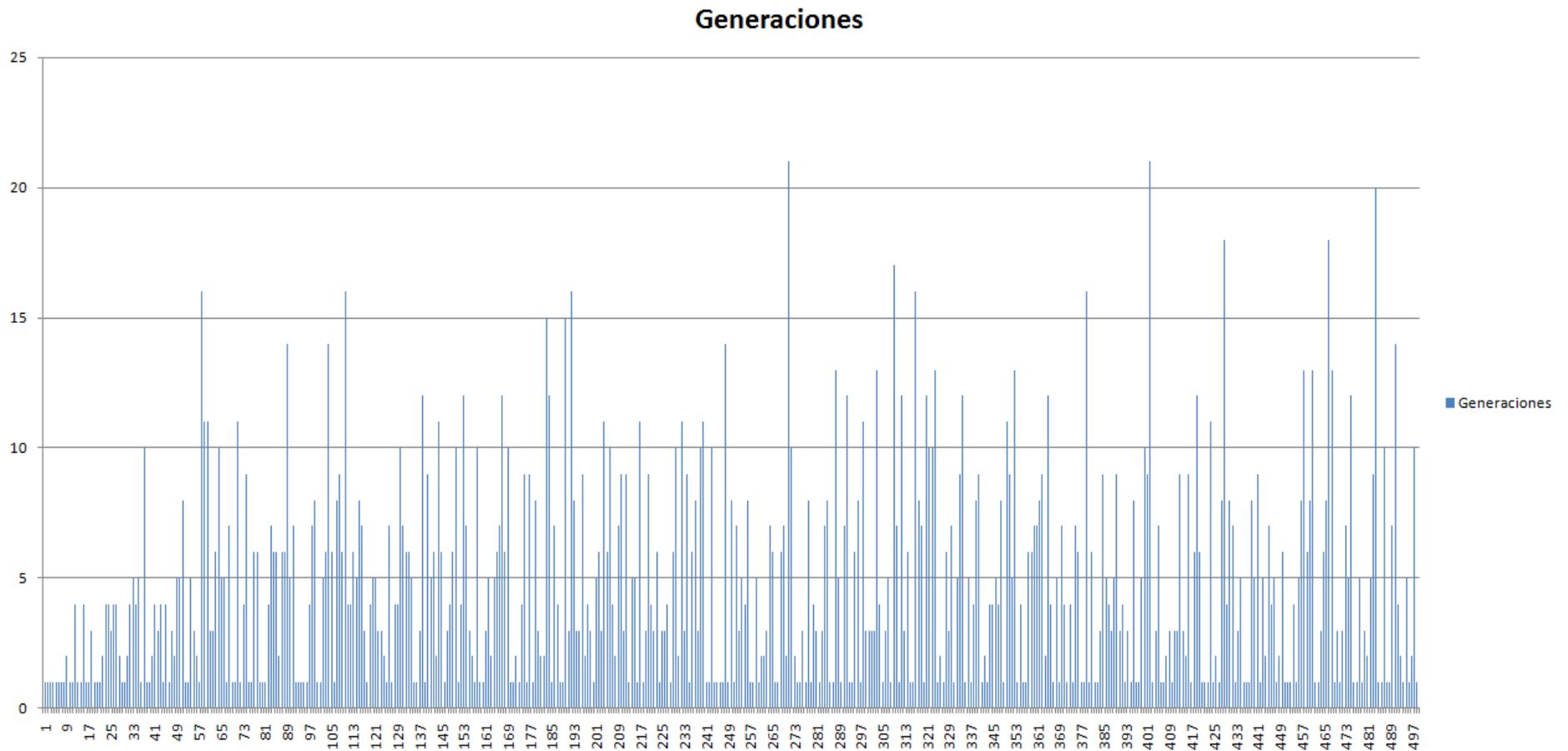


- Grafica de evolución frente a entrenadores pre generados.





- Grafica de coste generacional por entrenador.





5.4.10 Entrenador 8

- 20 mejores manos.

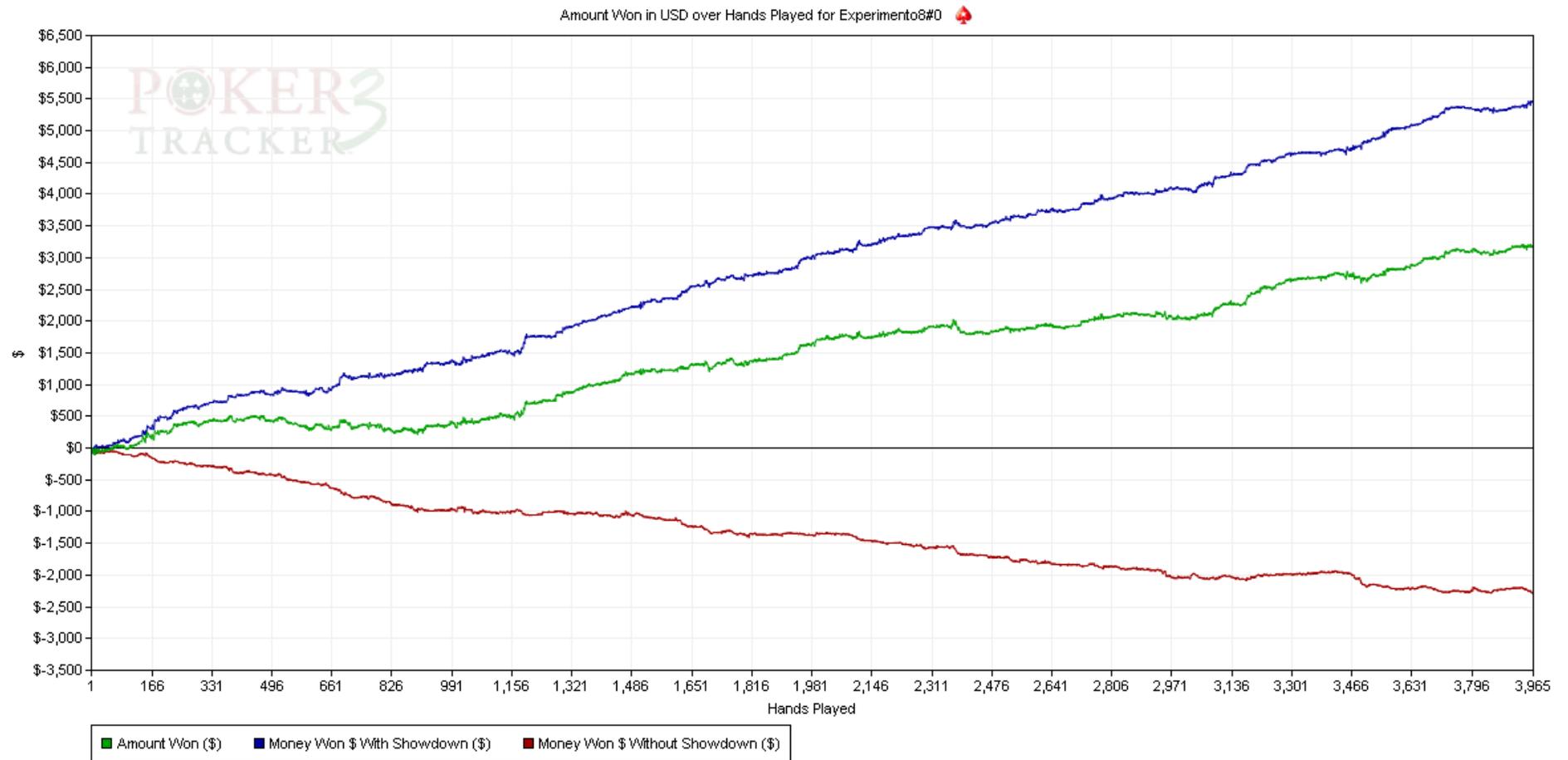
Hand	Times	Win %	Amount Won
QQ	99	68.69	\$954.25
AJs	81	60.49	\$942.50
KK	97	76.29	\$775.25
ATs	70	57.14	\$740.75
88	135	48.89	\$667.75
22	76	57.89	\$537.25
AKo	173	60.12	\$515.50
AQo	259	43.24	\$368.99
KQo	123	50.41	\$324.50
AJo	98	35.71	\$266.25
99	73	54.79	\$231.75
AA	29	68.97	\$227.00
KTo	97	49.48	\$207.50
33	48	45.83	\$187.75
66	83	49.40	\$161.75
JTs	28	50.00	\$141.25
J9o	39	43.59	\$129.25
KJs	31	45.16	\$115.50
44	58	31.03	\$111.25
A7o	22	77.27	\$102.00

- Combinaciones.

Final Hand	Times	Win %	Amount Won
Straight Flush	2	100.00	\$117.00
Four of a Kind	11	90.91	\$314.75
Full House	122	76.23	\$1,842.37
Flush	79	70.89	\$1,118.62
Straight	141	80.14	\$1,720.50
Three of a Kind	294	66.67	\$2,641.25
Two Pair	773	46.83	\$1,016.86
One Pair	1,732	36.84	-\$2,588.22
High Card	811	19.98	-\$3,006.50

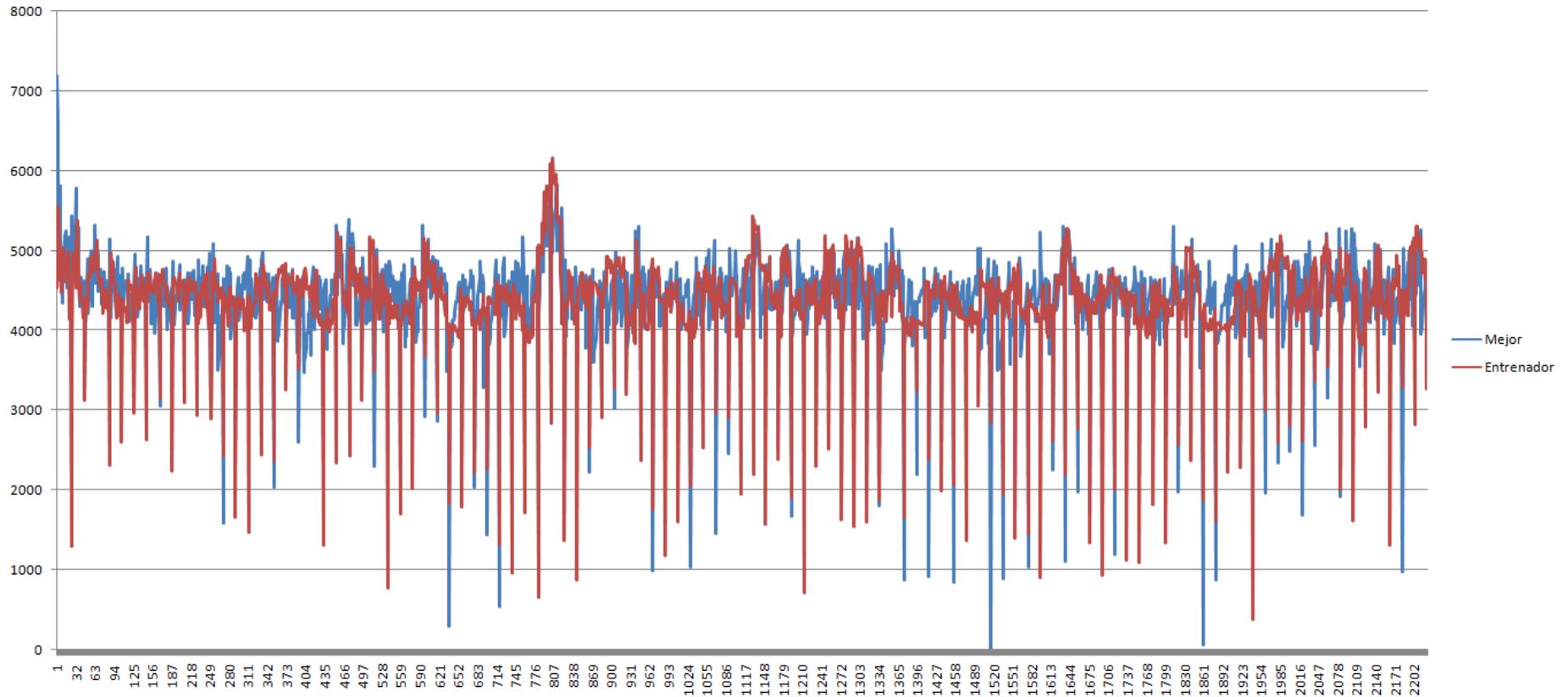


- Grafica de ganancias en la última generación.



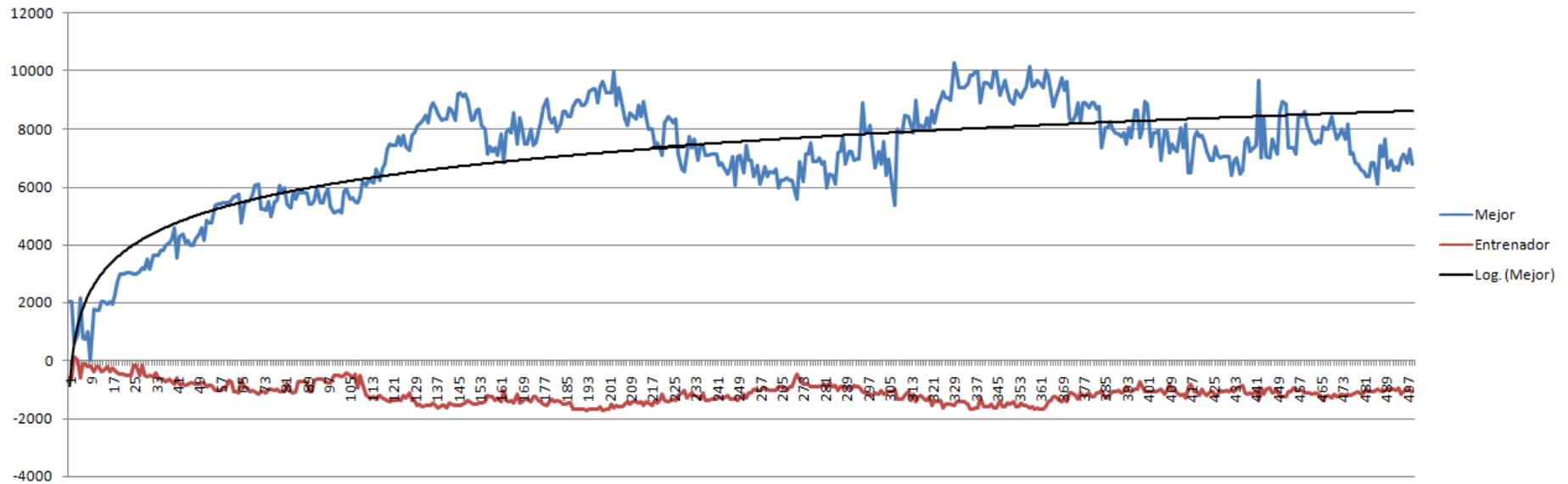


- Grafica de ganancias durante la ejecución.



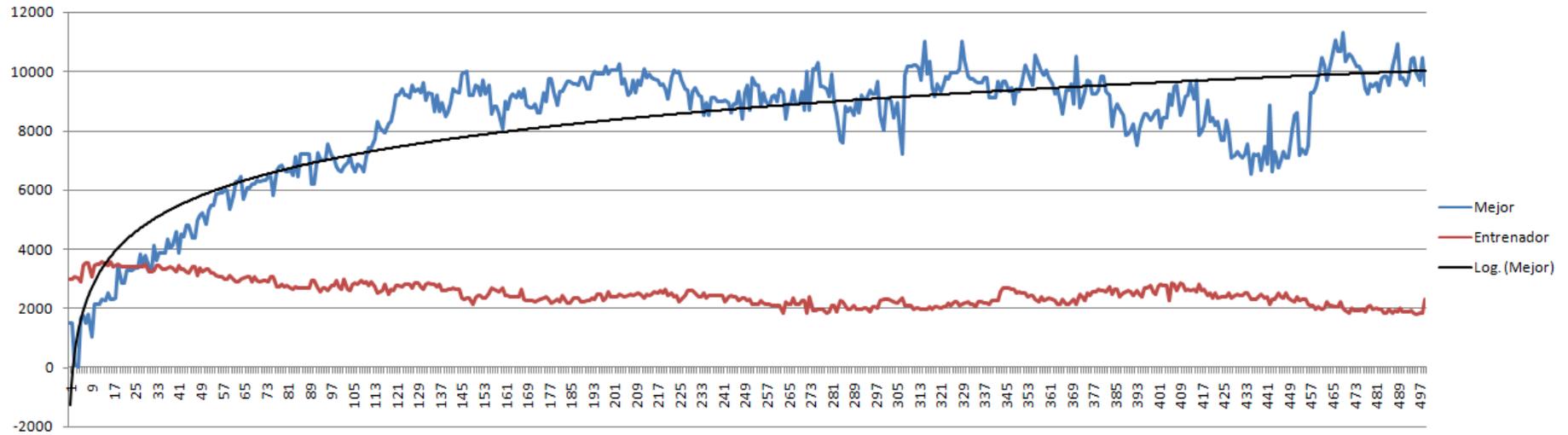


- Grafica de evolución frente a entrenadores aleatorios



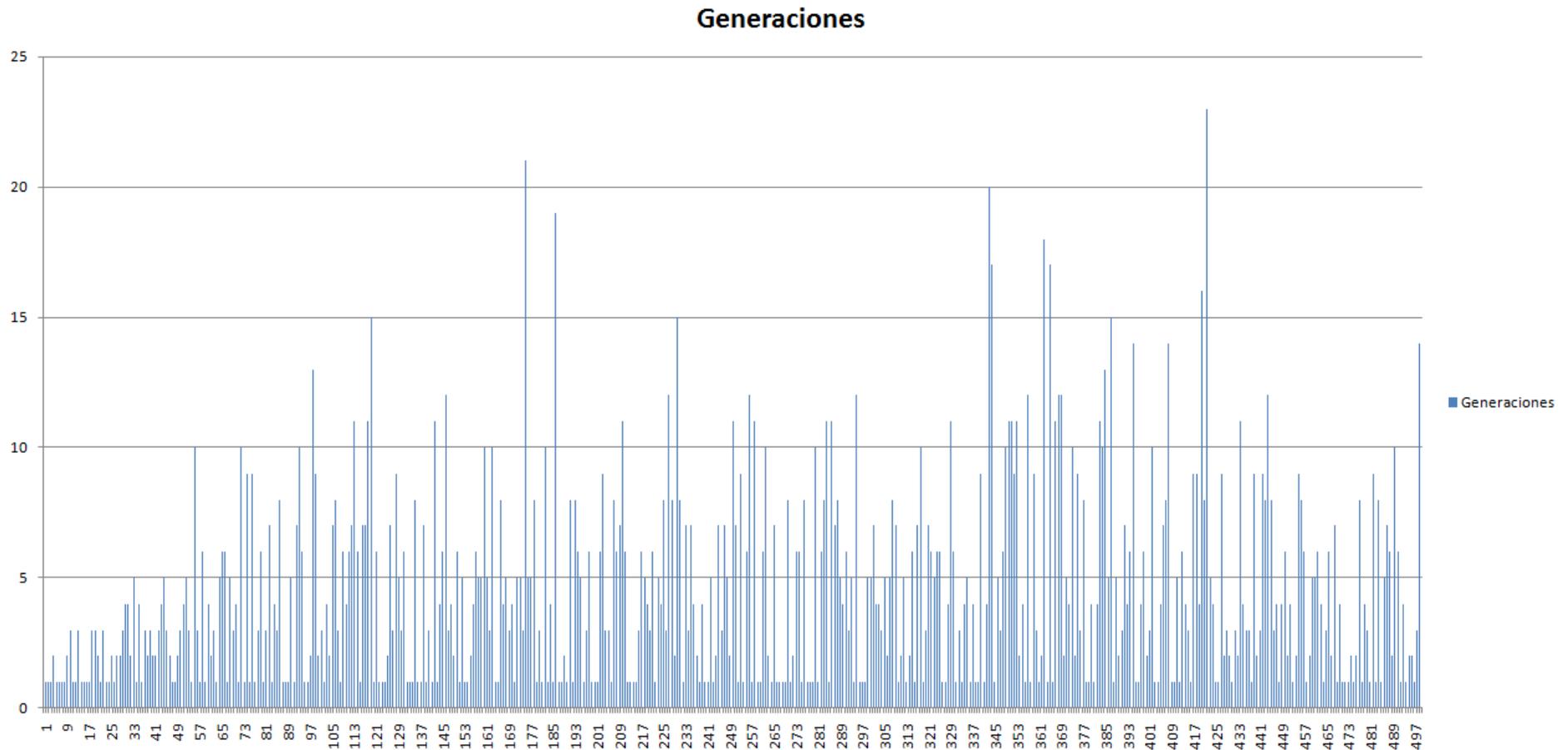


- Grafica de evolución frente a entrenadores pre generados.





- Grafica de coste generacional por entrenador.





5.4.11 Individuos contra entrenadores

En este apartado se describe la prueba de generación de individuos a partir de los entrenadores generados previamente.

La función de evaluación para esta prueba no co-evolucionan junto con la población, ya que se considera que el conjunto de entrenadores ya es suficientemente bueno de partida y no es necesario que este conjunto se modifique a lo largo de la ejecución.

Esta prueba se detallara con las mismas graficas que en las pruebas anteriores quitando la grafica de coste generacional, ya que debido a que la función de evaluación no co-evolucionan junto a la población.

Sin embargo, se ha incluido una grafica nueva, mostrando la evolución del individuo frente a los entrenadores.

- **Grafica de evolución frente entrenadores finales:** Esta grafica muestra la evolución de los individuos de la población (verde) frente al conjunto de entrenadores (rojo). El eje de las x representa el numero de generación y el eje de las y representa las ganancias de los jugadores.

A continuación se muestran las tablas de los resultados obtenidos en la generación del individuo final.

- 20 mejores manos.



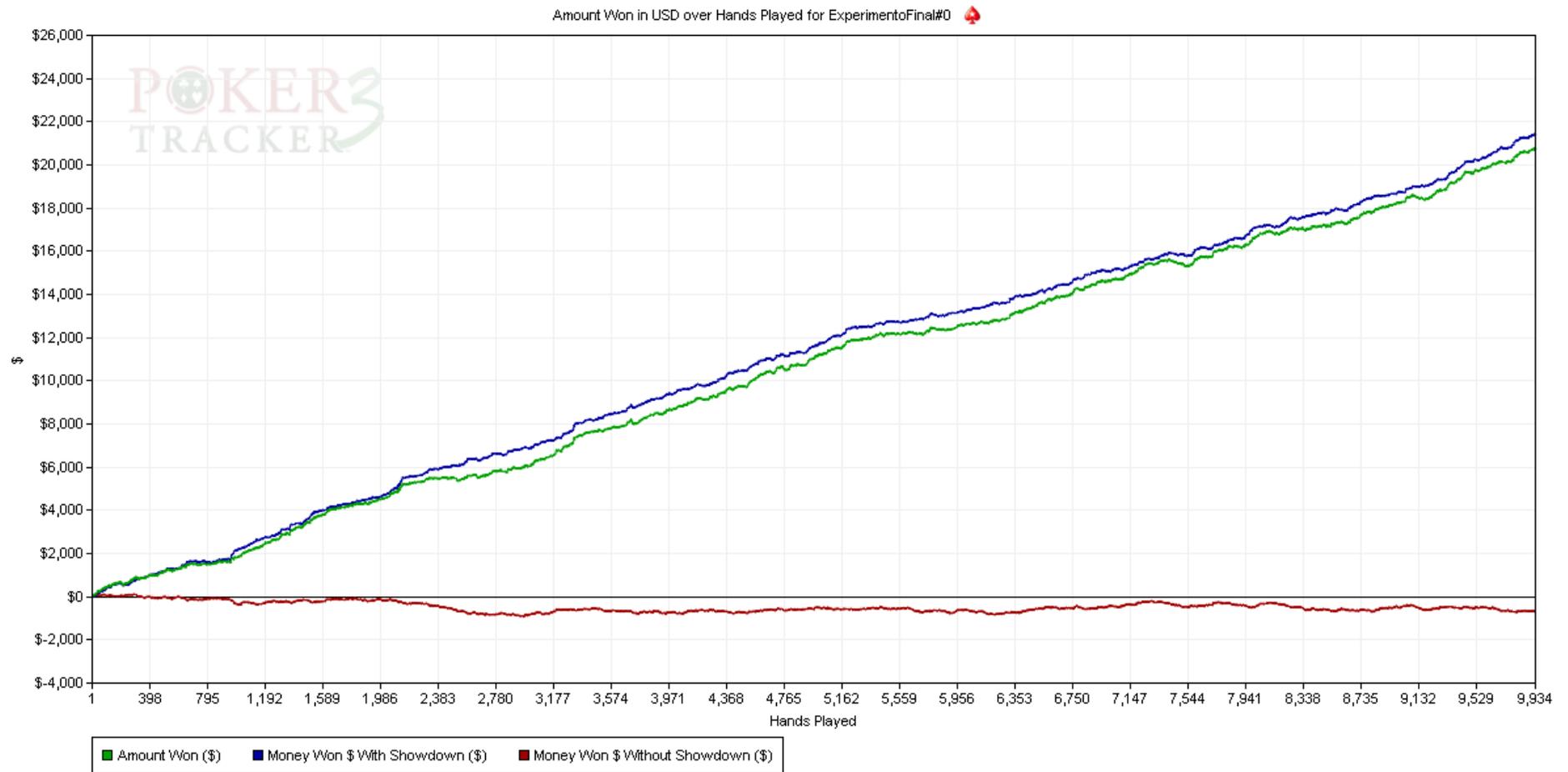
Hand	Times	Win %	Amount Won
AQo	493	57.61	\$2,394.93
ATo	538	47.96	\$1,666.61
77	231	57.58	\$1,637.25
AQs	196	58.67	\$1,594.00
AA	179	82.68	\$1,589.00
KQs	180	53.33	\$1,507.00
99	212	53.30	\$1,299.00
TT	161	67.70	\$1,283.25
KK	143	74.13	\$1,146.00
AKo	324	51.23	\$1,123.11
QQ	161	59.63	\$982.75
KJo	305	46.56	\$774.50
55	149	49.66	\$750.25
AJo	532	45.49	\$691.93
KQo	390	39.74	\$665.25
JJ	243	52.26	\$503.73
66	199	45.73	\$369.25
T9o	166	38.55	\$330.87
A8s	57	43.86	\$265.75
KTs	68	58.82	\$235.75
A6s	32	65.63	\$228.00

- Combinaciones.

Final Hand	Times	Win %	Amount Won
Straight Flush	15	93.33	\$341.25
Four of a Kind	41	100.00	\$952.50
Full House	306	83.33	\$5,024.47
Flush	212	75.00	\$2,256.25
Straight	369	72.63	\$3,062.82
Three of a Kind	523	68.45	\$5,134.12
Two Pair	2,492	54.90	\$10,733.74
One Pair	4,256	39.22	-\$1,689.14
High Card	1,720	18.66	-\$5,057.52

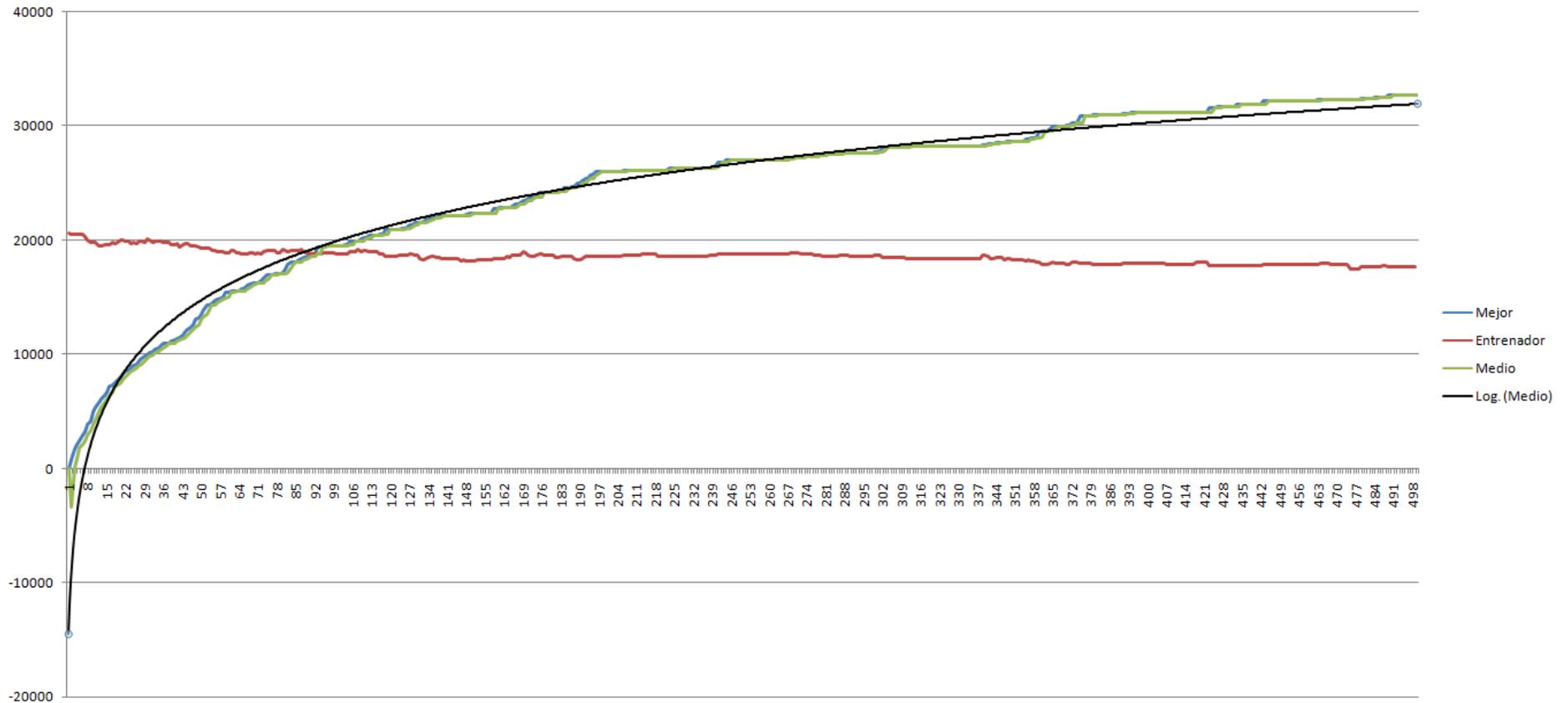


- Grafica de ganancias en la última generación.



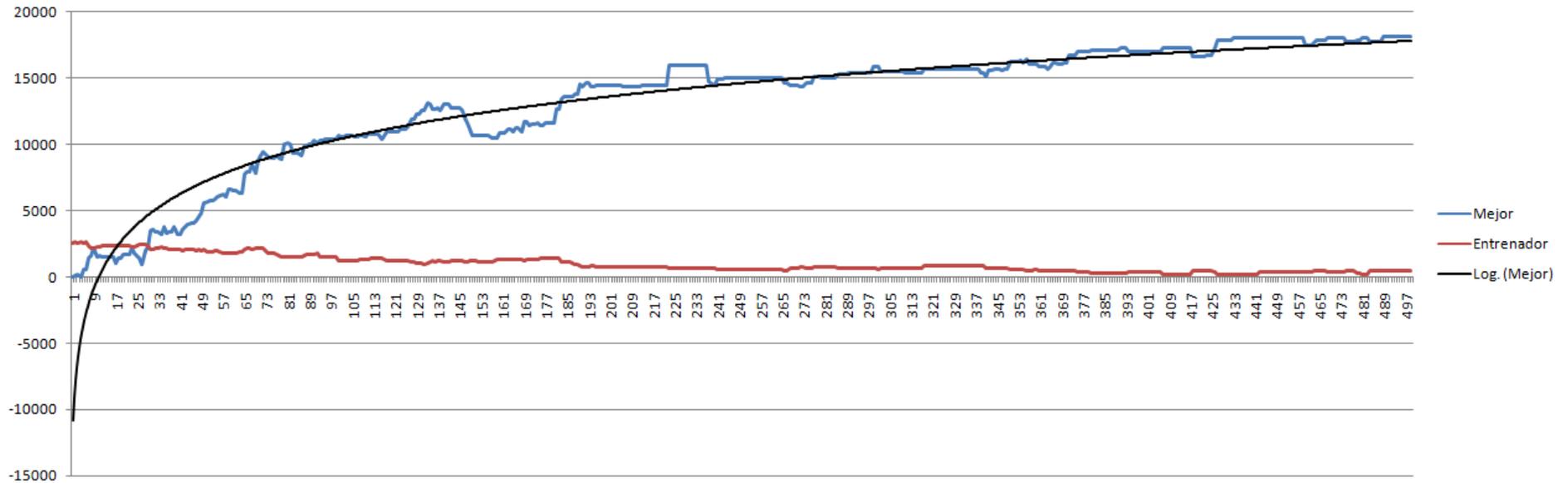


- Grafica de evolución frente entrenadores finales.



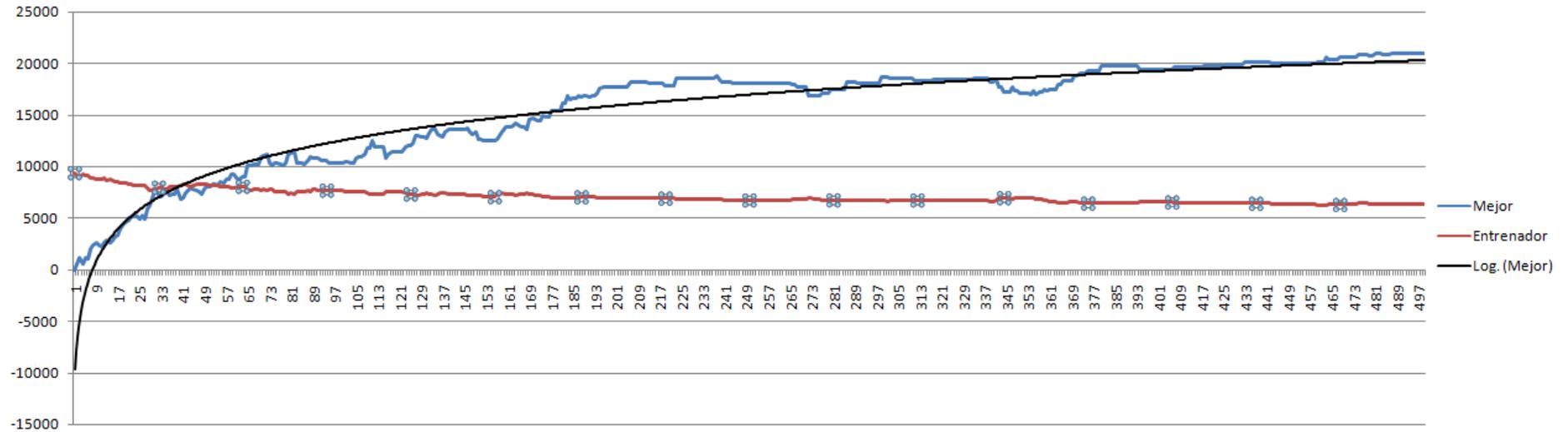


- Grafica de evolución frente a entrenadores aleatorios.





- Grafica de evolución frente a entrenadores pre generados.





5.5 Estudio del jugador final

En este apartado analizaremos los resultados del entrenamiento del jugador final. Utilizando una serie de indicadores procedentes del análisis de juego generado con la herramienta PokerTracker 3 definiremos que tipo de jugador es (Gonzalo García-Pelayo, Diego Pradera, Óscar García-Pelayo).

Los indicadores que utilizaremos son los siguientes:

- VP\$IP: Esta estadística indica el porcentaje de veces que un jugador ha visto voluntariamente el flop, es decir, el porcentaje de veces que un jugador ve el flop teniendo que poner dinero voluntariamente (la posición de ciego grande no cuentan para la extracción de dicho porcentaje).

Esta tabla muestra la clasificación en función del indicador.

VP\$IP	Clasificación
>40%	loose
(25%, 40%)	semi-loose
<25%	tigth

- PFR%: Esta estadística indica el porcentaje de veces que un jugador sube en el preflop.

Esta tabla muestra la clasificación en función de este indicador.

PFR%	Clasificación
$\geq 7\%$	agresivo
$< 7\%$	pasivo

- AF: Esta estadística el número de veces que sube la apuesta entre el número de veces que iguala.

Esta tabla muestra la clasificación en función de este indicador.

AF	Clasificación
≥ 2	agresivo
< 2	pasivo

En función de los indicadores descritos, se puede elaborar una clasificación de jugadores diferentes. A continuación se muestra una tabla en la que están clasificados diferentes tipos de jugadores en función de los indicadores descritos.

La tabla que se muestra a continuación muestra los distintos tipos de jugadores ordenados de peor a mejor calidad en términos de juego.



	VP\$IP	PFR%	AF
Pez	loose	pasivo	pasivo
Telefono	semi-loose	pasivo	pasivo
Dado	loose	agresivo	pasivo
Maniaco	loose	agresivo	agresivo
Ratón	tigth	pasivo	pasivo
Elefante	loose	pasivo	agresivo
Bomba	semi-loose	agresivo	pasivo
Roca	tigth	pasivo	agresivo
Triste	semi-loose	agresivo	pasivo
Alegre	semi-loose	agresivo	agresivo
Peligroso	tigth	agresivo	pasivo
Tiburón	tigth	agresivo	agresivo

Gracias a la herramienta de análisis de póker Pokertracker 3, se han podido obtener los siguientes datos sobre el juego del jugador final.

La siguiente tabla muestra diferentes indicadores sobre cómo juega el jugador final, entre ellos están los indicadores descritos.

Limit	VP\$IP	VP\$IP SB	PFR	W\$WSF	WTSD	W\$SD	AF
\$2/\$4	21.91	27.71	9.07	44.83	54.43	53.41	1.34

Después de analizar los indicadores obtenemos que nuestro jugador sigue la clasificación de ratón, un jugador tigth-pasivo-pasivo.

El jugador resultante sigue esta clasificación debido a que no necesita mejorar su juego más, ya que con esta tendencia de juego le basta para ganar al resto de entrenadores. La tendencia de juego pasivo del jugador generado, se debe a la tendencia de juego pasiva de los entrenadores. Este perfil no es suficiente para ganar a un jugador humano medio, por lo tanto el entrenamiento del jugador final no basta con utilizar un conjunto de entrenadores auto-generados. Sería interesante en cualquier caso llevar a cabo una validación con humanos para conseguir especificar mejor el perfil del jugador final, en un entorno de juego más realista.



6. Presupuesto

En esta sección se detallara la planificación llevada a cabo para desarrollar el proyecto. Se explicaran las fases en las que se divide el proyecto, así como el coste asociado.

Se hará un estudio del coste de recursos humanos y de materiales necesarios para la realización de este proyecto.

6.1 Planificación

El presente proyecto ha seguido una planificación temporal y económica, con el fin de realizarlo de la manera menos costosa, tanto en tiempo como en recursos y así minimizar los costes del proyecto.

La planificación del proyecto se ha dividido en las siguientes fases:

- **Estudio de viabilidad del sistema:** En esta fase se realiza un estudio de la viabilidad del sistema que se va a desarrollar. Se hace un estudio de todos los conocimientos y conceptos necesarios para el desarrollo del proyecto y se analiza si es viable llevarlo a cabo.
- **Diseño del sistema:** En esta fase se lleva a cabo el diseño del sistema que se va a desarrollar. Se llevara a cabo un diseño detallado de todos los subsistemas involucrados en el proyecto.
- **Implementación del sistema:** En esta fase se lleva a cabo la implementación del sistema que se ha diseñado previamente.
- **Pruebas:** En esta fase se realizan pruebas del sistema implementado para realizar un análisis de los resultados y de la calidad de las soluciones que genera.
- **Realización de la documentación:** Esta fase se lleva a cabo de forma simultánea al resto de fases. Durante la documentación se realizan todos los apuntes y se plasma en documentos todos los detalles de la realización del proyecto.



6.2 Materiales

En esta sección se detallara que recursos materiales y de software se han utilizado para el desarrollo del proyecto, así como el coste que ha implicado su uso.

6.2.1 Hardware

En esta sección se describe que recursos hardware se han utilizado para el desarrollo del proyecto y cuál ha sido su coste.

Material	Cantidad	Precio unidad (€)	Usó(Horas)	Tiempo Amortización (horas)	Coste Amortización (€/hora)	Precio (€)
Pc	1	600	168	34944	0,01717033	2,884615385
Subtotal	1	600			0,01717033	2,884615385

6.2.2 Software

En esta sección se describe que recursos software se han utilizado para el desarrollo del proyecto y cuál ha sido su coste.

Software	Cantidad	Precio unidad (€)	Usó(Horas)	Tiempo Amortización (horas)	Coste Amortización (€/hora)	Precio (€)
Pokertracker 3	1	90	103	34944	0,002575549	0,265281593
Eclipse	1	0	103	34944	0	0
Sistema operativo	1	150	103	34944	0,004292582	0,442135989
Service pack	1	100	103	34944	0,002861722	0,294757326
Subtotal	4	340			0,009729853	1,002174908

6.3 Recursos humanos

En esta sección se detallara que recursos humanos se han necesitado para el desarrollo del proyecto, así como el tiempo y el coste económico asociado a cada miembro del personal involucrado.



6.3.1 Diseño

En esta sección se describe el coste temporal y económico del diseño del sistema.

Tarea	Tiempo (horas)	Precio unidad (€)
Estudio de viabilidad del sistema	20	500
Planificación del proyecto	20	500
Identificación de subsistemas	25	625
Diseño de subsistemas	50	1250
Diseño de la entrada/salida	10	250
Subtotal	125	3125

6.3.2 Implementación

En esta sección se describe el coste temporal y económico de la implementación del sistema.

Tarea	Tiempo (horas)	Precio (€)
Implementación del algoritmo genético	10	100
Implementación del sistema de reglas	20	200
Implementación del parser para Pokertracker3	15	150
Implementación de la mano	5	50
Implementación de la salida de estadísticas	5	50
Subtotal	55	550



6.3.3 Pruebas

En esta sección se describe el coste temporal y económico de las pruebas que se han realizado con el sistema.

Tarea	Tiempo (horas)	Tiempo total(horas)	Cantidad	Precio unidad (€)	Precio (€)
Pruebas de consistencia	2	2	1	30	30
Pruebas de complejidad	2	2	1	30	30
Pruebas de Entrada/Salida	2	2	1	30	30
Pruebas del sistema	2	2	1	30	30
Prueba de generación de entrenadores	4	36	9	60	540
Prueba de generación de individuos	4	4	1	60	60
Subtotal	16	48	14	240	720

6.3.4 Documentación

En esta sección se describe el coste temporal y económico de la redacción de la documentación asociada al sistema.

Tarea	Tiempo (horas)	Precio unidad (€)
Recuperación de información	15	150
Redaccion de la memoria del proyecto	50	500
Análisis de resultados	10	100
Subtotal	65	975

6.3.5 Coste de personal

En esta sección se describe el coste del personal implicado en el desarrollo de este proyecto.

Puesto	Precio (€/hora)
Diseñador	25
Programador	10
Analista	15
Redactor	10



6.3.6 Total

En esta sección se describe el coste total de todos los recursos implicados en el desarrollo del proyecto.

Tarea	Total	Horas
Diseño	3125	125
Programacion	550	55
Pruebas	720	48
Documentación	975	65
Subtotal	5370	293
Materiales		
Harware	2,884615	
Software	1,002175	
Subtotal	3,88679	
Total	5373,887	



7. Conclusiones y trabajos futuros

En esta sección se abordarán las conclusiones que se han sacado de la realización de este trabajo, así como los trabajos futuros que se pueden realizar con los conocimientos y experiencia extraídos de este proyecto.

El trabajo ha resultado ser muy didáctico ya que se han adquirido muchos conocimientos nuevos para poder realizarlo, no solo en el campo del póker y de los algoritmos genéticos, sino que también en la organización y planificación de proyectos y muchos de los conocimientos adquiridos durante el grado en ingeniería informática.

La realización de este proyecto ha implicado la puesta en práctica de muchos conceptos y técnicas aprendidas a lo largo de la carrera, como los conocimientos sobre Procesadores del Lenguaje, modelos de concurrencia de Arquitectura de Computadores, Estructuras de Datos y Algoritmos e Ingeniería del Software. Sin los conocimientos obtenidos durante el grado no hubiera sido posible abordar este proyecto.

Adicionalmente ha resultado muy interesante orientar el trabajo de fin de grado hacia una labor de investigación sobre un tema tan actual como el póker, un tema de gran interés social, tratándose a su vez de un problema realmente complejo, tanto como problema de combinatoria, como a la hora de recabar información en una situación concreta, la información obtenida es incompleta.

El resultado obtenido de la realización de este sistema ha sido satisfactoria, ya que se han obtenido los resultados esperados, así como el cumplimiento de todos los objetivos propuestos para la finalización del proyecto. Si bien es cierto que la calidad de los jugadores generados no es mejor que la calidad de un jugador de póker humano medio, se ha superado el reto de obtener jugadores que jueguen de manera coherente sin la introducción de conocimiento de manera directa.

Adicionalmente se ha probado el sistema de co-evolución como un sistema útil para la resolución de problemas en los que se necesitan individuos con los que comparar una solución y de los que no se dispone a priori, y que su elaboración manual pueda resultar muy complicada.

Por tanto se concluye que el trabajo ha sido un éxito a todos los niveles, tanto como experiencia de trabajo y como puesta en práctica de conocimientos, conceptos y técnicas aprendidos. También ha sido un éxito personal ya que ha sido fuente de gran satisfacción y auto-realización ya que ha supuesto la prueba de las capacidades aprendidas durante la carrera que este proyecto pone fin.

Para generar una mejor solución obtenida se podrían ampliar el número de dimensiones contempladas por la codificación propuesta (por ejemplo número de jugadores que quedan en la mesa o subidas en cada una de las fases). También es cierto que se necesitarían conjuntos de manos más grandes, y tiempos de ejecución más largos para realizar esto. Adicionalmente, se podrían aplicar técnicas evolutivas multi-objetivo para



la generación de jugadores automáticos, siguiendo la línea conceptual del sistema desarrollado.

En un futuro, si los recursos hardware han mejorado, se podría abordar este problema utilizando mas grados de dimensión, y sistemas que implican un consumo de recursos más altos. También se podrían utilizar conjuntos de entrenadores más grandes e irlos rotando durante la ejecución. Así mismo también se podrían ponderar las ganancias y las pérdidas para desarrollar diferentes tipos de jugadores.

El principal problema con el que se ha lidiado a lo largo del proyecto, ha sido la falta de recursos hardware (procesador y en menor medida memoria), ya que las pruebas consumen gran cantidad de memoria y tiempo de ejecución. Por esto mismo, si se avanza suficientemente en la tecnología hardware en un futuro no muy lejano, se podría volver a abordar el problema con el mismo sistema pero utilizando un mayor número de dimensiones.



8. Bibliografía

"Computational Complexity and the Genetic Algorithm" Bart Rylander. University of Idaho 2001.

"Time complexity of genetic algorithms on exponentially scaled problems" Fernando G. Lobo, David E. Goldberg, Martin Pelikan. Universidade do Algarve, University of Illinois 2001.

"ALGORITHMS AND ASSESSMENT IN COMPUTER POKER" Darse Billings, University of Alberta 2006.

"Poker VIP" Jan Meinert 2010.

"Aprende a jugar al póquer en internet con los Pelayos" Gonzalo García-Pelayo, Diego Pradera, Óscar García-Pelayo 2009.

"Guía fácil del limit hold'em" Byron Jacobs 2010

"Building a No Limit Texas Hold'em Poker Agent Based on Game Logs Using Supervised Learning", Luis Felipe Teófilo y Luis Paulo Reis, Lecture Notes in Computer Science, Vol. 6752, pp. 73-82. 2011

"The Theory of Poker: A Professional Poker Player Teaches You How To Think Like One", David Sklansky, Cuarta edición, Two plus Two Publishing, 1994

"Hold'em Poker", David Sklansky, Two plus Two Publishing, 1997

"Hold'em Poker: For Advanced Players", David Sklansky and Mason Malmuth, Tercera edición, Two plus Two Publishing, 1999.

"Limit Hold'em: Winning Short-Handed Strategies", Terry Borer, Lawrence Mak y Barry Tanenbaum. D & B Publishing, 2009

"Evolution Strategies: A Comprehensive Introduction", H.-G. Beyer y H.-P. Schwefel. Journal Natural Computing, Vol. 1, no. 1, pp. 3-52, 2002.

"Function optimization with coevolutionary algorithms", F. Serebinski, A.Y. Zomaya, y P. Bouvry, Proceedings of the 03 International Intelligent Information Processing and Web Mining Conference, pp. 13-22, Springer, 2003.

"Epistasis for real encoding in genetic algorithms", S. Rochet, M. Slimaney y G. Venturini. Proceedings of the IEEE Australian and New Zealand Conference on Intelligent Information Systems, pp. 268-271. 1996

"PRACTICAL GENETIC ALGORITHMS" Randy L. Haupt, Sue Ellen Haupt 2004.

"Genetic algorithms in search, optimization, and machine learning", D.E. Goldberg, Addison-Wesley, 1989



“A genetic algorithm application in bankruptcy prediction modeling”. K. Shin y Y. Lee. *Expert Systems with Applications*, Vol. 23, no. 3, pp 321-328. 2002.

“Learning classifier systems”, P. Lanzi. En *Proceedings of the 11th Annual Conference Companion on Genetic and Evolutionary Computation Conference: Late Breaking Papers*, pp. 2853-2878. ACM. 2009.

“The challenge of poker”, D. Billings, A. Davidson, J. Schaeffer, y D. Szafron. *Artificial Intelligence*, Vol. 134, no. 1-2, pp.201-240, 2002.

“Learning to play strong Poker”, J. Schaeffer, D. Billings, L. Peña, and D. Szafron. En *The International Conference on Machine Learning Workshop on Game Playing*. J. Stefan Institute, 1999.

“Approximating game-theoretic optimal strategies for fullscale poker. D. Billings, N. Burch, A. Davidson, T. Schauenberg, R. Holte, J. Schaeffer, y D. Szafron. En *The Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence, IJCAI'03*, pp 661-668, 2003.

“A competitive Texas Hold'em poker player via automated abstraction and real-time equilibrium computation”, A. Gilpin y T. Sandholm. En *American Association of Artificial Intelligence National Conference, AAAI'06*, pp 1007-1013, 2006.

D. Billings, A. Davidson, T. Schauenberg, N. Burch, M. Bowling, R. Holte, J. Schaeffer, and D. Szafron. “Game-tree search with adaptation in stochastic imperfect-information games. In H. J. van den Herik, Y. Björnsson, y N. Netanyahu (editores), *4th International Conference on Computers and Games. Lecture Notes in Computer Science Vol. 3846*, pp. 21-34. Springer-Verlag, 2004

“Phil Gordon's Little Green Book: Lessons and Teachings in No Limit Texas Hold'em”, Phil Gordon, Gallery Books, 2005.

“Harrington on Hold 'em Expert Strategy for No Limit Tournaments, Vol. 1: Strategic Play”, Dan Harrington y Bill Robertie, Two Plus Two Publishing, 2004.

“Doyle Brunson's Super System: A Course in Power Poker”, Doyle 'Texas Dolly' Brunson, Chip Reese, Joey Hawthorne, Bobby Baldwin, Mike Caro y Dave Sklansky. Tercera edición, Cardoza Publishing, 2002.

Barricelli, Nils Aall (1963). "Numerical testing of evolution theories. Part II. Preliminary tests of performance, symbiogenesis and terrestrial life". *Acta Biotheoretica* (16): 99–126.

"On the Origin of Species" Charles Darwin 1859.