Introducción

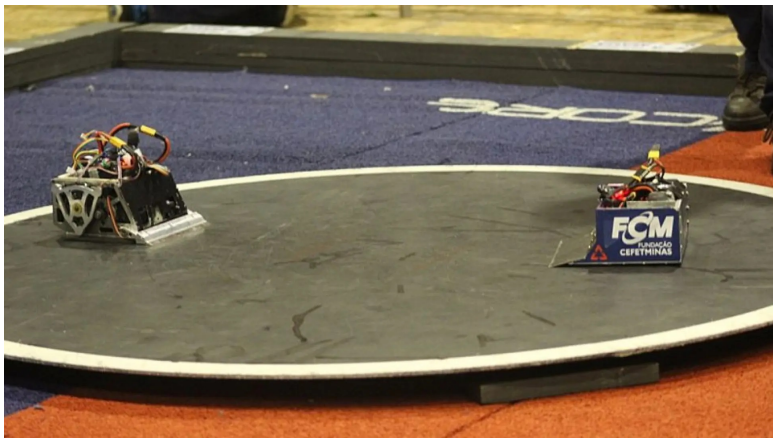

Figura 1: Robot Sumo

El "Robot Sumo" es un deporte real, que incluso se juega a nivel profesional. Análogo al Sumo entre humanos, este consiste en sacar al oponente del campo, solamente empujándolo. A veces se juega con roboces autónomos, y a veces a control remoto. En este trabajo se presenta una simulación de lo que se podría considerar una generalización de este deporte en su modalidad autónoma. En particular: en lugar de haber solo dos roboces, hay varios equipos de roboces que se enfrentan todos contra todos, y la forma del campo puede variar durante la partida, lo que generalmente se traduce en que a partir de cierto momento, el campo comienza a reducirse, hasta que desaparece. De esta simulación se sacarán conclusiones acerca de qué aptitudes físicas y "mentales"(o sea, de la IA) son más importantes y/o mejores para ganar una partida.

Detalles de la simulación

Por simplicidad, la simulación se realiza en 2D. Para simular la física (colisiones, fuerzas, etc.) se utiliza el motor de físicas Box2D. Para la visualización se usa la biblioteca Raylib. Toda acción que realiza un robot (atacar, posicionarse, rotar, etc.) se simula siempre aplicando fuerzas físicas sobre el mismo. Los robots se represetan como conjuntos de polígonos, y pueden tener forma arbitraria. Estos poseen además otras propiedades físicas, entre ellas: masa, restitución, fuerza (separada en fuerza linear y rotacional), precisión, rango de visión y frecuencia de procesado (cuantas veces por segundo analiza la situación actual para decidir qué hacer). Hay distintas "personalidades" de robot disponibles, cada una con sus propias preferencias y prioridades a la hora de jugar. Los robots pueden moverse libremente por el campo, pero para hallar su camino de un lugar a otro (esquivando obstáculos)

utiliza una discretización (idealmente fina) del campo, en forma de grafo. Respecto al campo, los parámetros más importantes con los que se puede experimentar son el radio, a los cuántos segundos comienza a a reducirse, y a qué velocidad se reduce. El código de los jugadores, su cerebro, y la lógica del campo se pueden encontrar respectivamente en fighter.py, brain.py y scene.py. Se presenta una visualización de la batalla, que se puede desactivar para que la simulación vaya más rápido (útil a la hora de sacar estadísticas). Con la configuración con la que realizamos los experimentos, 30 frames representan un segundo dentro de la simulación(aunque esto es un parámetro configurable). Esto no significa que la simulación vaya a 30 frames por segundo constantes, sino que cada vez que pasan 30 frames, pasa un segundo de la simulación. Así, si el programa va a 210 fps, a cada segundo de la vida real están pasando 7 segundos dentro de la simulación.

La arquitectura de los agentes

Para la inteligencia de los agentes (los jugadores) se decidió optar por una arquitectura BDI. Los "Belief" serían todos los datos que tiene el agente para tomar decisiones, que mayormente corresponde a las propiedades del agente, y de los agentes que están en su campo de visión, además de sus posiciones y velocidades. Los "Desires" son planes relativamente abstractos, como "Atacando al agente X" o "Ejecutando la fase 2 del ataque X". Los "Intentions" corresponden a qué se debe hacer específicamente para ejecutar esos planes, como por ejemplo "Cargando hacia la posición (x,y)", o "Rotando en el lugar", o "Frenando".

Detalles de la inteligencia de los agentes

Lo primero que hace un agente es inferir qué está haciendo el resto de los agentes dentro de su campo de visión (atacándolo, huyendo de él, etc.). Esto se logra mediante un modelo oculto de Markov (con memoria) por cada agente dentro del campo de visión, cuya entrada son la percepción del agente. Una vez un jugador infiere los estados del resto, queda tomar decisiones al respecto.

Con el objetivo principal de iterar rápidamente cuando se está creando un comportamiento de la IA, se diseñó un sistema a alto nivel. Su uso es el siguiente: se crea una instancia de la clase Personality, y se le añaden reglas. Estas reglas consisten en implicaciones, del tipo: si se cumplen estas condiciones, ejecutar secuencialmente esta serie de acciones (cada una de las acciones se ejecuta hasta que se cumple una condición, pasando así a la siguiente acción en la cola). Estas acciones a ejecutar pueden ser incondicionales (como atacar a un enemigo, o alejarse del borde) o pueden ser llamados a otras reglas (y las reglas tienen condición). Esto permite, por ejemplo, usar recursividad, con una regla que ejecute una serie de acciones incondicionales, y luego se llame a si misma. Para decidir por qué regla optar cuando se cumplen las condiciones de varias, se puede especificar un "score" por cada regla, que puede depender de variables, como la cantidad de enemigos que ve el agente; a mayor score, mayor probabilidad de que se elija una regla.

Con el sistema descrito se pueden crear inteligencias arbitrariamente complejas, y se dan como ejemplo principalmente 3 IAs:

-IA estándar: intenta llegar a un balance entre ataque y defensa. Se arriesga, pero no mucho. Su ataque es el estándar (cargar directamente contra el enemigo, rotando).
-IA precavida/paranoica: prioriza la defensa. Trata de mantenerse muy alejada del borde del campo. Si hay muchos jugadores con una probabilidad relativamente alta de estarla atacando, huye, o empieza a rotar descontroladamente para alejarlos. Su ataque es el estándar.
-IA planificadora: Prefiere un ataque más complejo, de dos fases: se aleja para tomar impulso, y luego carga contra aquel al que decidió atacar.

Hay dos comportamientos que sí tienen en común las 3 opciones: si están muy cerca del borde del campo (que significa "muy cerca" lo define cada una), priorizar alejarse, y si no hay nadie en su radio de visión, explorar.

Experimentos

A continuación se describen (y dan resultados de) algunos de los experimentos realizados en la simulación. En los experimentos, los agentes usan la IA estándar, a no ser que se diga lo contrario.

Acerca de las distintas IAs de los agentes

¿Cuál de las IAs que proponemos para los agentes es más efectiva? o equivalentemente ¿Qué forma de jugar es más efectiva? Es difícil hacer un ranking exacto de las IAs, ya que su efectividad varía de acuerdo al tamaño del campo, o algunos atributos físicos del robot (precisión o campo de visión por ejemplo). Así que fijemos un campo de batalla estándar (10 unidades de radio, 20 segundos antes de que el campo se empiece a achicar, y el radio del campo de achicca a 2 unidades por segundo), y unos equipos estándar (sus propiedades exactas se encuentran en el fichero de python que se usó para este experimento), en este caso idénticos en todo sentido (excepto que los robots de cada equipo los controla una IA distinta), y veamos los resultados.

La clara ganadora es la IA más precavida. Esto se debe mayormente a que la regla de "el campo comienza a reducirse hasta que desaparece" convierte a

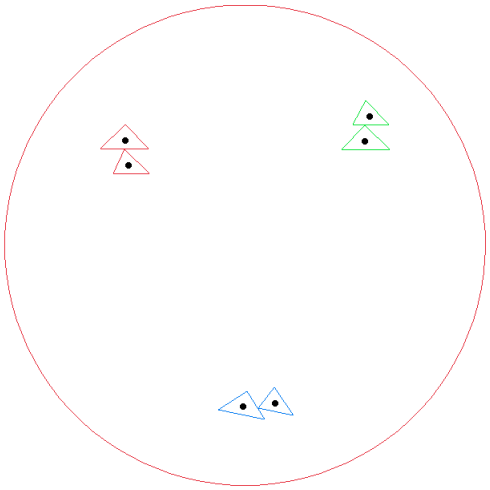

Figura 2: El campo para este experimento. Los equipos Rojo (R), Verde (G) y Azul(B) tienen las IAs estándar, paranóica, y planificadora respectivamente.

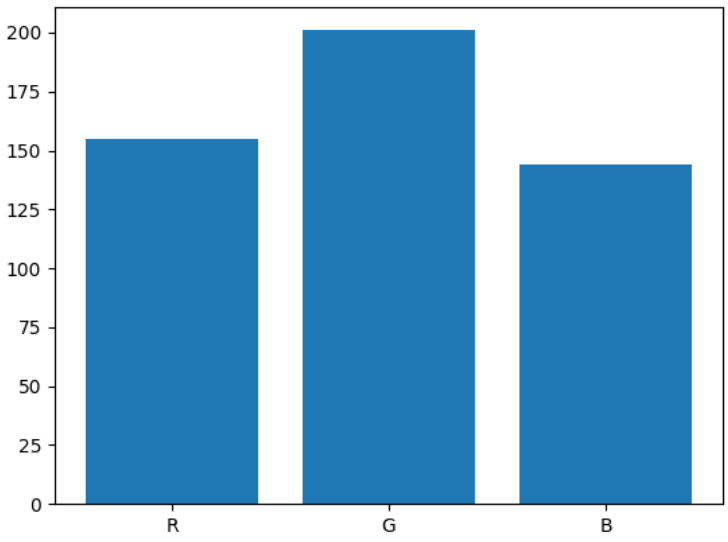

Figura 3: Los resultados del experimento, para 500 simulaciones. La IA paranóica (precavida) gana. Las entre las otras dos hay diferencia también, pero no realmente significativa.

este juego en uno de supervivencia, lo que premia las tácticas defensivas y el control del centro del campo. Los resultados para las otras dos son no presentan una diferencia tan marcada entre sí, pero una posible explicación para ella es que en un juego rápido, los movimientos compuestos se demoran demasiado, y resultan penalizados (entoces el ataque de dos fases de la IA planificadora sería sub-óptimo).

Acerca de las formas de los robots

¿Importa la forma de los robots? Volvamos a tomar un campo estandar y equipos con propiedades idénticas (ahora salvo la forma), y veamos:

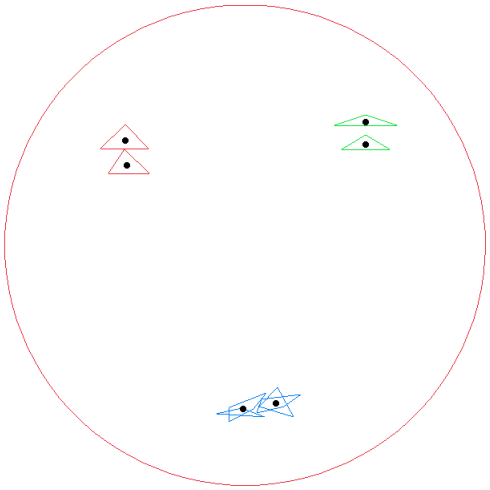

Figura 4: El campo para este experimento. Los equipos Rojo (R), Verde (G) y Azul(B) tienen forma normal, alargada y compuesta/cóncava respectivamente.

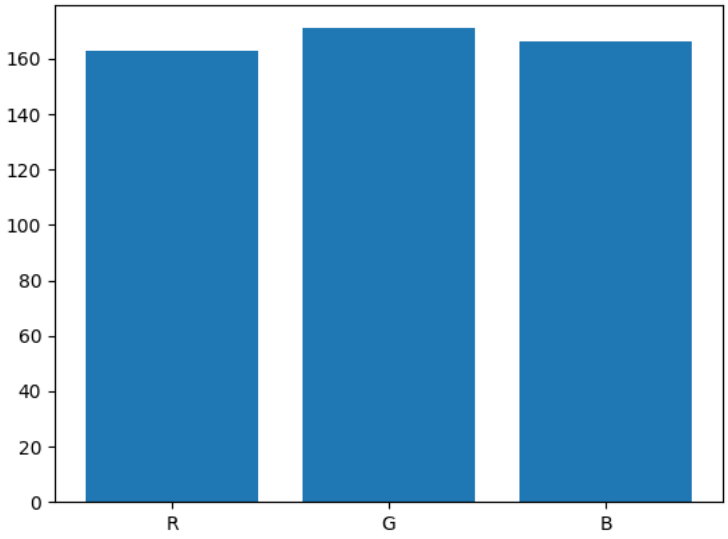

Figura 5: 500 simulaciones. La diferencia es despreciable

Resulta que no hay mucha diferencia de rendimiento entre las formas de los robots, siempre y cuando tengan tamaños similares.

Acerca de los tamaños de los robots

Bueno, entonces... ¿Y el tamaño? Veamos:

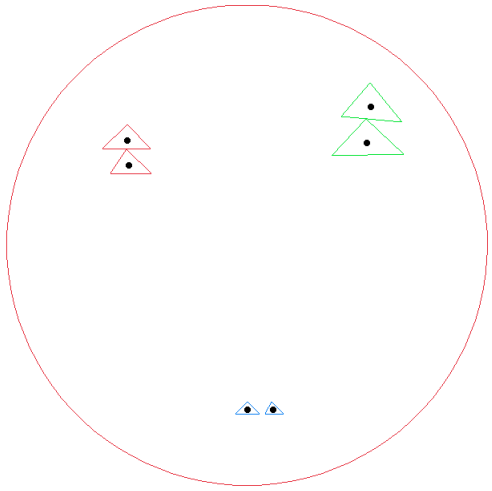

Figura 6: El campo para este experimento. Los equipos Rojo (R), Verde (G) y Azul(B) son de tamaño normal, grande y pequeño respectivamente. La forma de los robocitos son iguales entre los equipos.

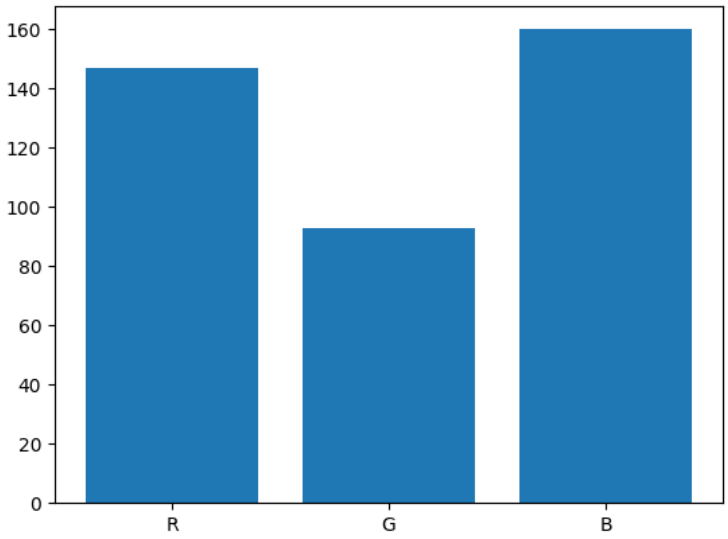

Figura 7: 500 simulaciones. Los robots grandes son muy poco efectivos. Los pequeños son los más efectivos, pero solo un poquito más que los medianos.

Pues los tamaños sí importan. Los robots grandes son extremadamente poco efectivos. Esto no es porque tengan más áreas para ser considerada "fuera del campo", ya que según las reglas de la simulación, un robot se considera fuera del campo solo cuando su centro de masas lo está, no cuando cualquier pedacito suyo lo esté. Una posible explicación para esta diferencia tan grande es que (como se observa en simulaciones individuales) los robots grandes tienen más superficie para ser golpeados por otros, y en particular por varios otros robots a la vez; que los golpeen varios a la vez por el mismo lado es equivalente a que un robot muy fuerte los golpeen, lo que los pone en desventaja. De hecho, los robots así de grandes suelen perder de primeros, cuando todavía hay muchos otros en el campo.

Acerca de las posiciones iniciales

¿Importa la posición inicial de los equipos en su rendimiento? En particular ¿Qué sucede si un equipo empieza ya en el centro? El centro es importante. Veamos, con equipos completamente idénticos:

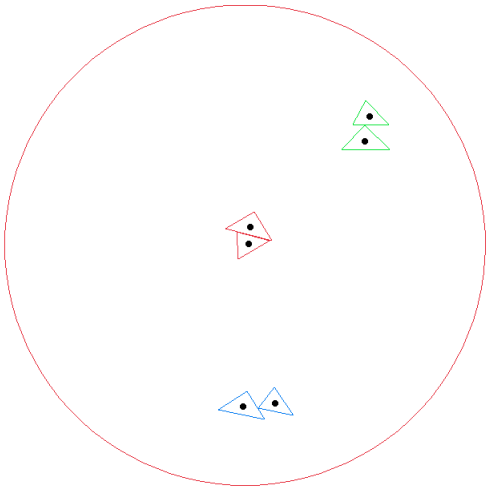

Figura 8: El campo para este experimento.

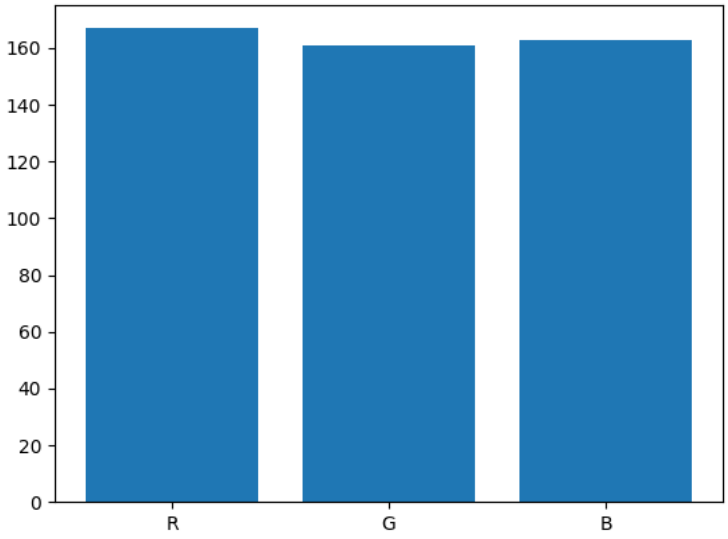

Figura 9: 500 simulaciones. No hay difecencia significativa

No hay diferencia que valga. Empezar en el centro no aporta ni ventaja ni desventaja estratégica. Esto tiene sentido, porque es rápido llegar al centro, el campo es pequeño; además, el estado inicial se diluye cuando transcurre la partida.

Acerca del resto de los parámetros físicos

Además de las características 'inteligentes' de los agentes, también su efectividad en el combate se ve afectada por la configuración de ciertos otros parámetros físicos que poseen. En este apartado se realiza un análisis acerca del comportamiento de dichos parámetros en función de determinadas variables independientes.

Cabe destacar que el experimento realizado para realizar el análisis se llevo a cabo de una forma peculiar; y es que en nuestro proyecto se abordo la problemática de encontrar una configuración de parámetros para un equipo de robots que le permitieran obtener un desempeño notable contra otros dos equipos fijos. Para resolver este problema se implemento un algoritmo evolutivo que recibía como argumentos las configuraciones de los dos equipos contrarios y devolvía la configuración del equipo ganador, así como el historial completo de la evolución. En este contexto se escogió como data para responder a las interrogantes del comportamiento de los parámetros físicos al historial de evolución del algoritmo genético para 5 escenarios diferentes.

El primer escenario que se le presentó al algoritmo genético estaba compuesto por dos equipos bastante equilibrados en cuanto a sus parámetros, y en los escenarios posteriores se fueron incorporando a los mejores equipos encontrados por el algoritmo genético.

En cada uno de los 5 escenarios diferentes se consideraron 10 generaciones compuestas las mismas por 10 individuos (equipos de 2 robots). Además, en cada generación a cada individuo se le determinó su rendimiento promedio basándose en los resultados de 10 simulaciones. Esto da un total de unos 1000 individuos evaluados (unos 500 equipos) y unas 5000 simulaciones.

A continuación se realizarán observaciones acerca de los resultados obtenidos. No obstante en la carpeta 'simulation results' se encuentran 5 archivos JSON con las configuraciones de los mejores equipos encontrados en cada escenario y 5 con cada uno de los historiales de evolución. También existe un subdirectorio 'plots' con un total de 27 graficas obtenidas sobre esa data, de entre las cuales presentaremos aquí las que consideramos más significativas.

Los parámetros físicos analizados son los siguientes:
-Densidad de masa [1-15]
-Restitución (0-8]
-Fuerza linear [50-500]
-Fuerza de rotación [10-100]

-Precisión (0-1)
-Radio de visión [2-20]
-Frecuencia de reacción [1-10]
Al lado de cada parámetro se muestra el rango de valores que puede tomar (estas limitaciones se imponen para mantener cierto realismo). Es necesario señalar algo importante en el caso de las fuerzas; y es que en un inicio se populó al algoritmo genético con individuos para los cuales los valores de dichos parámetros distribuían uniformemente de manera aleatoria en los intervalos descritos, pero rápidamente se observo que si los valores se alejaban demasiado de un valor central el desempeño de los robots se hacía en general inútil (pues se desplazaban excesivamente rápido y por ende salían del ring). Por este motivo se decidió popular al algoritmo genético con valores de estos parámetros que siguieran una distribución normal con media 150 y std $\frac{50}{3}$ para la fuerza lineal, y 45 y $\frac{55}{3}$ para la fuerza de rotación respectivamente.

Comenzaremos nuestro análisis respondiendo a la pregunta de cuáles de los parámetros tuvo un valor medio con un comportamiento más uniforme a medida que aumentaba el porcentaje de victorias. Los datos que se presentan a partir de este punto salieron de las 5000 simulaciones anteriormente descitas. En este sentido tenemos 3 parámetros que destacan sobre los demás, los cuales son: la fuerza de rotación, la masa y el rango de visibilidad. En el caso de la fuerza de rotación y el rango de visibilidad se aprecio un incremento bastante estable a partir del 20% de victorias (en ambos casos los equipos con un 90% de victorias presentaban una diferencia de alrededor de 3 puntos con respecto a los de un 20%). Si bien este incremento no parece significativo, si lo es el hecho de que las medias convergieran de manera tan uniforme dado el alto número de componentes aleatorios que existen en la simulación.

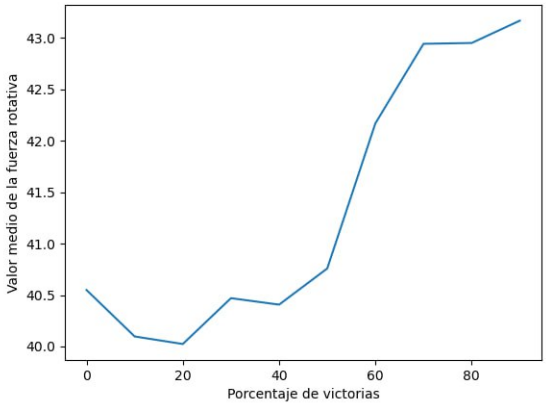

Figura 10: Valor medio de la fuerza rotacional

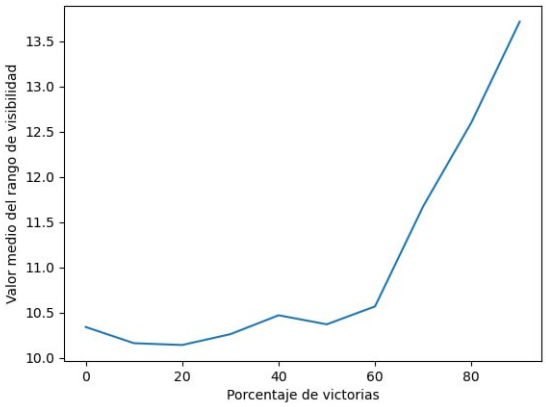

Figura 11: Valor medio del radio de visión

En el caso de la masa sucede lo contrario. Con un valor medio en general cercano a 7.5 (la mitad del intervalo en el cual toma valores), la media de este parámetro experimentó un decrecimiento sostenido a medida que se incrementaba el porcentaje de victorias. Más aun, la tasa de ese decrecimiento fue bastante similar hasta aproximadamente el 70% de victorias, para luego aumentar de manera exponencial; generando así una convergencia muy acelerada que permitió en un intervalo pequeño descender alrededor de 0.8 puntos (entre el 70% y el 90% de victorias).

Nuevamente aunque en un final se descendió poco más de una unidad solamente, se percibe muy claramente que para los escenarios presentados la optimalidad de las soluciones va de la mano con un decrecimiento en la masa.

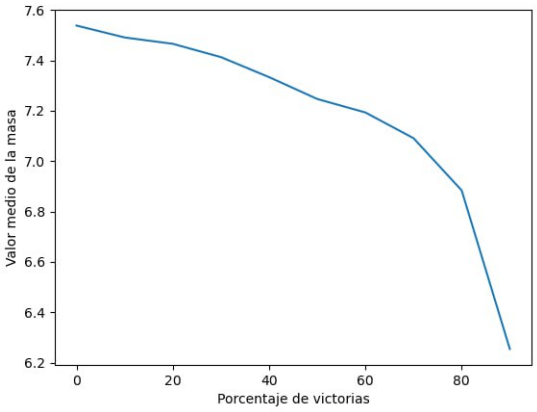

Figura 12: Valor medio de la masa

Otras magnitudes que presentan cierta estabilidad son la fuerza lineal y la frecuencia de actualización, al menos hasta un 70%-80% de victorias, en las cuales experimentan cambios abruptos de monotonía. Esto puede deberse a la naturaleza del algoritmo genético de en determinados momentos potenciar la exploración en lugar de la explotación, trasladando así su atención a otras regiones del espacio de búsqueda. Otra interrogante interesante después de observar el comportamiento de las medias de las magnitudes físicas de los robots es analizar como de dispersos estuvieron los valores obtenidos con respecto a dicha media. Con este propósito vamos a analizar el comportamiento de las desviaciones lineales estándar de los atributos, nuevamente, en función del porcentaje de victorias obtenidas por los equipos.

En este caso existe solo un atributo cuya std presenta un comportamiento bastante uniforme, y este es la precisión de los robots, cuya desviación estándar disminuye a medida que aumenta el porcentaje de victorias. Esta desviación estándar comienza siendo de un 22% aproximadamente y termina

en un 11% para aquellos equipos con más de un 90% de triunfos. Mas importante aun es el hecho de que no existen cambios abruptos en la monotonía de este decrecimiento, lo cual es un claro indicativo de que la precisión es un parámetro de peso a la hora de analizar la efectividad de los robots en el combate. A pesar de que un valor de precisión absurdamente alto no es un requerimiento (los mejores equipos de cada uno de los escenarios tienen en promedio un 72% de precisión y los equipos con más del 90% un 68% de precisión), sí que se cumple que la desviación estándar es pequeña (un 8% para el primer caso y un 12% para el segundo)

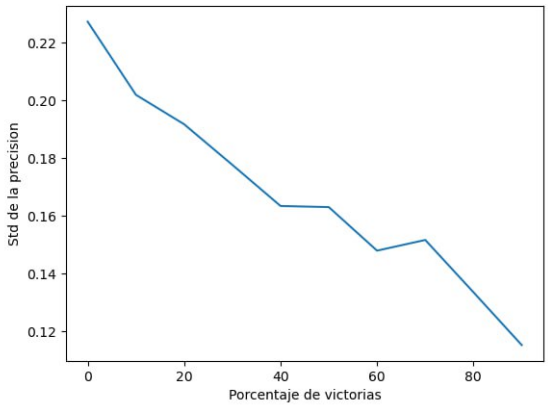

Figura 13: Desviación estándar de la precisión

Otras características, como la frecuencia de actualización, masa, fuerza rotativa y rango de visibilidad también presentan cierta uniformidad hasta llegar a valores relativamente elevados de victorias, en los cuales presentan cambios muy pronunciados; esto también tiene que ver con el hecho de que a partir del 60% de victorias la cantidad de equipos comienza a decrecer con una velocidad cercana a 2^n (por ejemplo, existen 82 equipos con más de un 60% de triunfos, 42 con más de un 70%, 15 con más de un 80% y 4 con más de un 90%) lo que genera que desaparezca mucha información referente a óptimos locales.

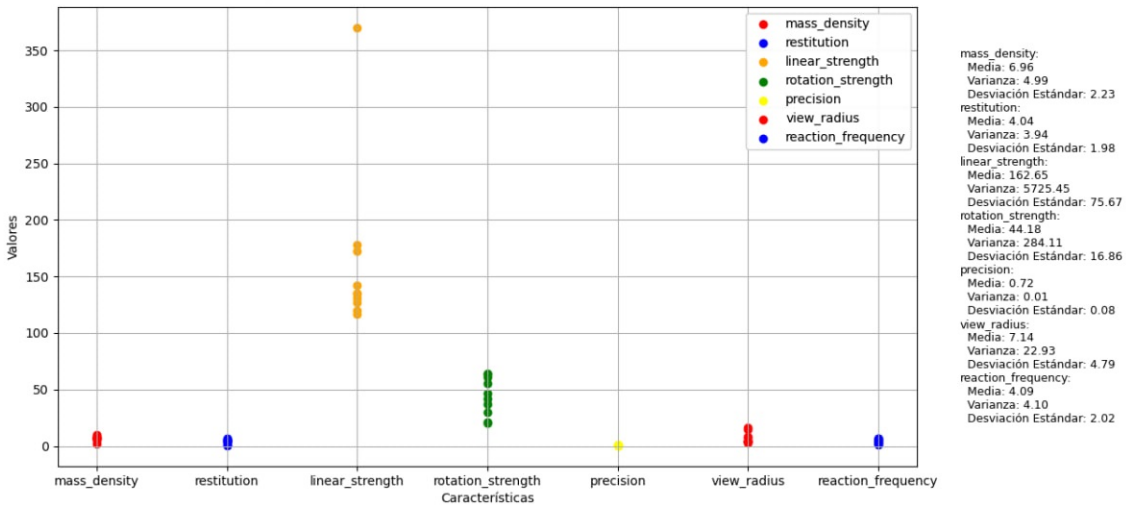

Figura 14: Dispersión de las carácterísticas de los equipos más óptimos en 5 escenarios diferentes

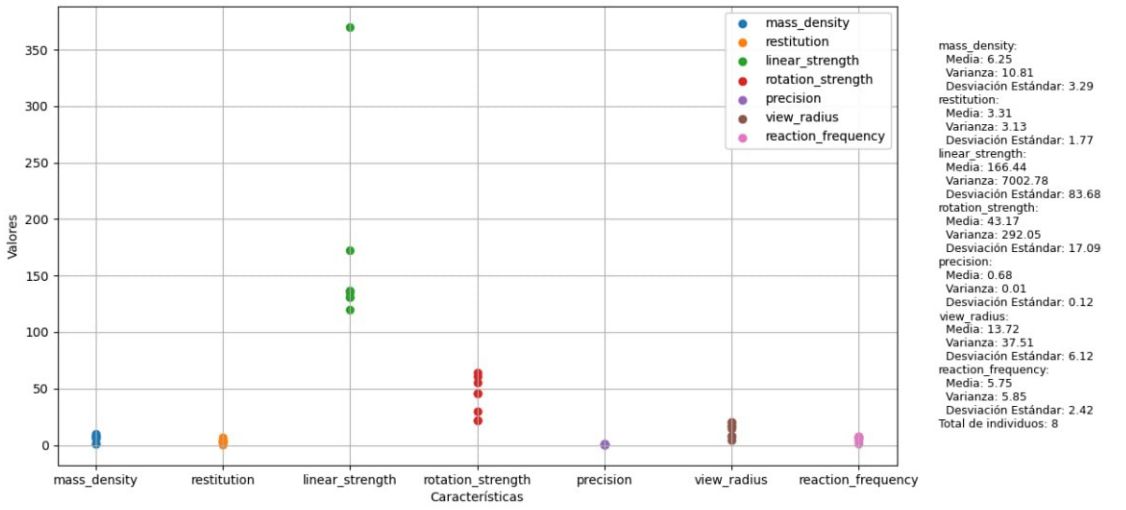

Figura 15: Dispersión para equipos con más de un 90% de victorias