Memoria

Jorge Vázquez Acevedo

Jaime Miguel Hernández

Contenido

[Problema a resolver 3](#_Toc7995960)

[Técnica para abordar el problema 4](#_Toc7995961)

[Método para adaptar la técnica al problema 5](#_Toc7995962)

[Planteamiento Inicial 7](#_Toc7995963)

[Problemas Iniciales y Soluciones 9](#_Toc7995964)

[Resultados Intermedios y Final 11](#_Toc7995965)

[Comentarios sobre el Resultado Final 12](#_Toc7995966)

[Qué hemos aprendido 13](#_Toc7995967)

# Problema a resolver

DIME SI ME CUELO O QUÉ PONER SI NO.

Mas que problema a resolver seria como “tema a tratar” o algo así. En realidad no estamos resolviendo un problema como tal, sino haciendo una investigación sobre un modelo evolutivo de simulación de criaturas.

El problema que vamos a plantear es ver qué estrategias acaban siendo las mejores a la hora de enfrentar a las criaturas entre sí. Éstas tienen una serie de acciones que se verán condicionadas en gran medida por sus características intrínsecas y en otra por los combates, es decir, por la criatura a la que se van a enfrentar. También queremos dar a conocer cuáles de las características llega a determinar el número de victorias de una o varias criaturas. Por no hablar también sobre el conocer en qué generación empiezan a converger los individuos, es decir, cuándo unos individuos empiezan a predominar sobre el resto de forma clara independientemente de las sucesivas generaciones.

-------------

El trabajo realizado se basa en una investigación sobre un modelo evolutivo en el que simulamos un ecosistema de criaturas, las cuales combaten para determinar cual es más fuerte. En esta investigación propondremos un modelo del ecosistema, con una herramienta para simular los combates, y utilizando un algoritmo evolutivo trataremos de encontrar criaturas cada vez mejores (de acorde al mencionado sistema de combate).

Durante el desarrollo de la investigación introduciremos cambios en el algoritmo evolutivo, que permitirá encontrar resultados diferentes, con una convergencia más rápida o con mayor diversidad, entre otros aspectos importantes. Así mismo, analizaremos que resultado tienen estos cambios, si el resultado es el que esperábamos cuando se pensó el cambio, y si mejora alguno de los aspectos mencionados: diversidad, velocidad de convergencia, exploración de otros mínimos relativos, “clusterización” de los individuos en el dominio de búsqueda.

# Técnica para abordar el problema / Conceptos previos: algoritmos genéticos

Es un poco lo que has escrito tú, pero ciñiendote más a una especie de “exposición” básica de los conceptos de algoritmos genéticos: por que usamos algoritmos genéticos y no fuerza bruta (o cualquier otro método de búsqueda), que ventajas tienen, que inconvenientes. Explicación sin entrar en muchos detalles de los conceptos de población, generación, selección, mutaciñon, cruce, operadores de cruce, operadores de mutación, función de evaluación… Es para que una persona que no sepa lo que son los algoritmos genéticos pueda tener una idea un poco general antes de ponernos a comentar “hemos cambiado el operador de cruce de esto a esto porque me permite mejor diversidad blablablá…”. Si les soltamos eso de primero no van a entender ni papa, y no es eso.

Antes de entrar en la investigación realizada, vamos a exponer brevemente algunos de los conceptos relacionados con los algoritmos genéticos y sus símiles con los procesos de evolución de la naturaleza.

En primer lugar, como su propio nombre indica, un algoritmo evolutivo es un algoritmo de búsqueda basado en el proceso natural de evolución de criaturas, y en el concepto de supervivencia del más fuerte. Permite encontrar soluciones, de manera más óptima que si se hiciera por fuerza brutal, mediante la adaptación del problema a un genoma (una manera de describir las soluciones), una función de evaluación (que permite, valga la redundancia, evaluar cuanto de buena es una solución), unos operadores de selección (cómo escogemos, dependiendo del valor obtenido en la función de evaluación, que criaturas van a perdurar en la siguiente generación), de cruce (la forma que tienen las soluciones de entremezclarse en su afán de mejorar), y de mutación (permitiendo a las soluciones salir de máximos relativos en su búsqueda de absolutos).

Quizá la diferencia más significativa en el uso que hacemos nosotros del algoritmo, con respecto a lo que ocurre en la naturaleza, es que trabajamos con generaciones en tiempo discreto. Es decir, las diferentes criaturas no van naciendo y muriendo de manera simultánea, sino que todas van pasando por las mismas fases al mismo tiempo. Por lo demás, conceptualmente hay muchísimas similitudes entre la evolución natural y los algoritmos evolutivos utilizados.

------

La técnica que vamos a aplicar son los algoritmos genéticos. Vamos a hacer un modelo de simulación. En concreto un modelo de control, más bien, queremos conocer al detalle cómo las criaturas acaban comportándose y qué características son las que más acaban desarrollándose. De ahí que por último acabemos por la visualización de los datos, para sacar las conclusiones. Sin embargo, también en gran parte podríamos decir que se trata de un modelo de optimización combinatoria, debido a que solo las mejores criaturas -las que obtienen un mayor número de victorias- son merecedoras de ser objeto de nuestro estudio. Aunque si uno ha de decantarse, sería la optimización quien se podría decir que es la más acertada. Dado lo descrito arriba en el apartado anterior, planteamos un problema relativamente abierto. Dentro del campo que nos ataña, pero reflejando la perfecta similitud y posible aplicación a otro problema de la realidad, o bien incluso virtual, como perfectamente podría ser para un videojuego. En concreto y por poner un ejemplo, a la hora de hacer un videojuego de lucha para poder tener una idea clara cómo hacer unos personajes equilibrados y no hacer uno que gana siempre al resto, perdiendo así el juego entretenimiento.

OPTIMIZACIÓN COMBINATORIA. VS CONTROL PREGUNTAR

HABLAR SOBRE LOS ALGORITMOS GENETICOS EN GENERAL?

Módulo de evaluación

Módulo de población

Módulo de reproducción

HABLAMOS DE ESTO DE LOS MÓDULOS AQUÍ O DESPUÉS??

## Genoma

Conjunto de datos que representa al individuo: que características va a tener, como se va a comportar… No hay una manera fija de expresar esta información, pudiendo ser, por ejemplo, una lista de valores enteros, valores en coma flotante, caracteres, información binaria, o incluso mezcla de todos ellos.

El proceso de definición del genoma representa un paso muy importante en el desarrollo del problema, ya que dependiendo de como se representen los individuos será más fácil posteriormente controlar las dinámicas de convergencia o clusterización de los individuos. Si el genoma no está correctamente definido pueden darse casos, por ejemplo, de sobreajuste (los individuos aprenden a superar esa función de evaluación concreta, pero si la cambias por otra similar no se comportan de manera adecuada), o de no llegar a encontrar una solución subóptima (una solución buena) al no adaptarse correctamente al problema en cuestión.

## Función de Evaluación

Se encarga de juzgar cuan bueno es un individuo (o cuan buena es la solución que representa el individuo). Al igual que el genoma, es muy importante una correcta definición, pues puede llevar también a casos de sobreajuste, si la función es demasiado específica; o de subajuste, si es una función demasiado genérica, y el genoma no puede “adaptarse”.

Generalmente se suele tener una función de evaluación fija, que se mantiene constante a lo largo de toda la ejecución del algoritmo (distancia recorrida por un robot, altura máxima alcanzada en un cohete, distancia al objetivo…). Sin embargo, puesto que en realidad lo que intentamos es juzgar cuan buena es una solución, la función de evaluación puede ser, como en nuestro caso, hacer un combate entre los individuos de la población y ver cuál obtiene más victorias. De esta manera, tenemos una función de evaluación dinámica, en la que en una generación una criatura puede ser muy buena (ganar al resto de sus competidores), y unas cuantas generaciones más adelante ser una criatura de las peores (no ganando a nadie).

## Operadores de selección

Los operadores de selección son, básicamente, la manera en la que seleccionamos los individuos de cara a la continuación en la siguiente generación. Haciendo un simil con la naturaleza, mientras la función de evaluación determinaría cuanto de saludable puede llegar a estar una especie, el operador de selección vendría a ser de que forma dos criaturas se ven “emparejadas” de cara a la reproducción (en el caso de criaturas con reproducción sexual). A lo largo de nuestra investigación utilizaremos dos tipos de selección: selección por ruleta y selección por torneo, cada una con sus ventajas e inconvenientes.

En la selección por ruleta a las criaturas se les asigna una probabilidad de ser elegidas para el cruce de acorde al valor obtenido en la función de evaluación. Cuanto mejor sea la criatura, más probabilidad tendrá de ser elegida. El problema de este tipo de selección es que una vez una criatura ha llegado a un máximo subóptimo y obtiene una gran probabilidad en la ruleta, se va a producir una convergencia muy fuerte a dicho individuo, negando la posibilidad de salir del mencionado máximo subóptimo.

La selección por torneo consiste en emparejar aleatoriamente las criaturas en un cuadro de eliminación, y en cada enfrentamiento quedarse con la criatura que tenga una mejor función de evaluación. Por ejemplo, con cuatro criaturas (A,B,C,D), se emparejaría aleatoriamente A con C y B con D, y nos quedamos con las que mejor valoración tengan (A, D). Entonces, estas dos criaturas son las que se acabaran cruzando para obtener el individuo de la siguiente generación. Lo positivo de este método es que da más oportunidades a los individuos con puntuaciones más bajas, aumentando la diversidad a costa de la velocidad de convergencia (si un individuo está en un máximo, le llevará más generaciones “imponer” su gen dominante).

## Operadores de cruce

Una vez hemos seleccionado dos criaturas para su “reproducción”, hay que definir como mezclamos los genes de ambas para obtener un nuevo individuo para la población siguiente. Esto depende en gran medida de la implementación del genoma de las criaturas, pero en general se pueden clasificar en los siguientes tipos:

* Cruce homogéneo: consiste en seleccionar aleatoriamente el gen de uno de los progenitores, y descartar el otro. Se usa principalmente con genes discretos; o bien enteros, o bien caracteres o bien valores booleanos.
* Cruce por “media”: se obtiene la media (no necesariamente aritmética, ya que puede estar ajustada con un multiplicador aleatorio para evitar que se acabe comprimiendo al cero) de los genes de ambas criaturas.
* Cruce en un punto/en dos puntos/ en N puntos: Aunque hemos considerado que carece de sentido para nuestro caso concreto, este operador vendría a ser cortar la cadena de números que representan el genoma en N+1 trozos, y mezclar los trozos enteros, en lugar de gen a gen como en el homogéneo.

## Operadores de mutación

Al igual que en la naturaleza pueden ocurrir mutaciones, tanto para bien como para mal, el algoritmo genético utiliza esta técnica para evitar que las soluciones representadas por los individuos caigan en un máximo relativo del espacio de búsqueda y no salgan de esa zona. Mediante la mutación, lo que conseguimos es “saltar” alejándote del máximo relativo en el que está atascada la solución, y llegar a otra solución que puede (o no) ser otro máximo relativo por el que diversificar la población.

# Adaptación del problema al algoritmo genético

Este título podría abarcar, de manera mas general, como hemos modelizado las criaturas, el combate, las poblaciones… Es describir como hemos “adaptado el problema de búsqueda” a cosas que podemos computar. Describimos como hemos hecho la función de evaluación, como funciona el genoma de las criaturas (los genes de características físicas y los genes de comportamiento…)… Puedes por ejemplo dividir este apartado en subapartados con cada una de las partes: Criatura/Genoma/Espacio de búsqueda/Valores prohibidos, Combate/Función evaluación…

También dividiría lo que es el modelo del problema (Criatura y combate) con el algoritmo genético en si (operadores de cruce y tal, que ya están explicados un poco en el apartado anterior).

En este apartado hablaremos de como hemos adaptado el problema (ver que criaturas son más fuertes) a los conceptos de algoritmos genéticos. Para ello, vamos a describir el modelo de combate que tenemos, así como el modelo de criaturas y que características las diferencian unas de otras.

En primer lugar, cada criatura tendrá cinco características físicas: aguante (HP), ataque (AT), armadura (AR), energía (EN) y agilidad (AG). La suma de todas ellas tiene que ser 30, para que de esta forma las criaturas estén equilibradas.

* HP: Determina la cantidad de puntos de vida con la que empiezan los combates.
* AT: La cantidad de puntos de daño que potencialmente pueden producir las criaturas con su ataque.
* AR: Proporcional a los puntos de daño absorbidos en cada ataque recibido.
* EN: Recurso necesario para poder realizar acciones en el combate; ya sea atacar, defenderse… En la sección de combate se explicará más detalladamente su uso.
* AG: En el caso de que la criatura opte por intentar esquivar un ataque, la agilidad marcará la probabilidad de que esto ocurra.

Además de las características físicas, las criaturas dispondrán también de un sistema que les permite, a partir de una serie de indicadores durante el combate (diferencia en puntos de vida actuales, diferencias en ataque…) tomar la decisión de que acción realizar en cada turno. Cada par acción-indicador tendrá una ponderación, y se tomará la acción con la suma de indicadores ponderados más alta.

Por lo tanto, el genoma lo representaremos como una lista de números, de los cuales los 30 primeros serán enteros representando la cualidad que mejoran (ejemplo más abajo), y los siguientes 20 serán las ponderaciones de las acciones a realizar.

Por otro lado, el combate entre dos criaturas funciona como sigue:

* Cada criatura empieza el combate con la cantidad de puntos de vida que tenga como HP.
* Los turnos son simultáneos. Es decir, las dos criaturas tienen que tomar la decisión sobre que acción realizar sin saber que va a hacer el otro.
* Existen cuatro posibles acciones en cada turno: Atacar, Bloquear, Esquivar o Pasar.
  + Atacar necesita dos puntos de energía, e intenta causar daño al rival; con el máximo siento el valor de AT.
  + Bloquear necesita tres puntos de energía, y asegura que independientemente de lo que haga el rival no recibe puntos de daño.
  + Esquivar intenta evitar el daño del rival. Necesita un punto de energía, en contraposición a bloquear que necesita 3. Sin embargo, la probabilidad de evitar el daño oscila entre 0.33 y 0.66, de acorde con la cantidad de Agilidad que tenga la criatura.
  + Pasar regenera un punto de energía, pero deja vulnerable a la criatura al daño.
* El combate acaba cuando una o ambas criaturas están con cero puntos de vida o menos. Entonces, la criatura con menos puntos de vida (incluyendo números negativos), pierde. En el caso de que ambas criaturas acaben el combate con los mismos puntos de vida se produce un empate.
* De cada combate se guarda la información no solo de quien ha salido victorioso, sino también del daño causado, recibido, y la diferencia de puntos de vida entre las dos criaturas.

En la función de evaluación no solo se tiene en cuenta el número de victorias de la criatura, sino también el daño total que ha causado y la suma de diferencias de vida al final de los combates. De acorde a estos valores, se le otorga una puntuación mediante esta fórmula: 10·NVictorias+2·DañoHecho+DifVida

-------

Ahora nos definiremos en la parte teórica de la computación evolutiva todos los campos en los que nos vamos a mover y necesitar. Para ello y una mejor compresión vamos a usar conceptos los conceptos vistos en clase sobre las componentes del modelo de optimización: DIME SI ESTA FORMA DE PLANTEARLO TE GUSTA O TE PARECE FEO.

\* Función Objetivo o de Coste: Basaremos en gran medida la calidad de cualquier solución mediante el número de victorias de los individuos. Esto no es del todo cierto, pero más adelante lo especificaremos con mayor profundidad.

\* Espacio de Búsqueda: Como no sabemos a qué se tienen que parecer o cómo deben ser nuestros individuos, este espacio de búsqueda es abierto a cualquier solución que nos conceda la computadora. Ahora adelante matizamos.

\* Restricciones, condiciones y valores prohibidos: Las características de los individuos estarán en rangos discretos y sus diferentes acciones estarán condiciones no solo por sus capacidades sino también por las del contrincante. Como valores prohibidos podemos destacar que no aceptamos que dos criaturas luchen cuando una de las dos tenga menos puntos de vida que el mínimo establecido. Todo esto entraremos en detalle más adelante en el Planteamiento Inicial.

\* Resolución: METAHURISTICAS, HEURISTIAS NO USAMOS NADA DE ESO NO??

\*Solución: Por la naturaleza del problema, todas nuestras soluciones serán subóptimas, pues ninguna necesariamente es la mejor, ya que cada población se genera de forma aleatoria y porque, recalcando, el problema y nuestro planteamiento no está diseñado para que se tienda a una solución. O si se da el caso, no es de nuestro conocimiento.

\*Objetivo Múltiple: Cabe recalcar que como la naturaleza del problema es abierta, depende qué pregunta nos hagamos o en qué campo nos queremos centrar para incluso rozar la idea de la objetivo único. Pero como decimo, es porque vamos a abordar todo, por lo que podemos decir que se trata de objetivo múltiple.

\*Dimensión del problema:

\*Grado de restricción: Serán todas las características de cada criatura.

\* Método de Evaluación: Utilizamos desviacion estandard de la criatura con respecto a la media de la poblacion para penalizar a las criaturas que esten cerca de otras mejores (con mayor número de victorias) pero parecidas (pues tenderían baja, y por ende su puntuacion sera peor)

Módulo de evaluación

\* Método de Población:

Se inicializa en cada ejecución de forma aleatoria

El tamaño de la población son menos de 30 individuos.

\* Método de selección: Por Ruleta o Por Torneo.

\*Mecanismo de Inserción: Nos basaremos en un modelo estacionario, ya que los nuevos individuos reemplazan a una serie de miembros de la población previa NO SEGURO AHORA DUDO

\*Módulo de Reproducción: PREGUNTAR A JORGE NO SEGURO, Nos basaremos en parejas de individuos, en las que cogeremos aleatoriamente características de cada progenitor. DÓNDE QUEDA this.ratioMutacion=4; this.tasaMutacion=0.5;

\*Genoma: Para nosotros será lo que llamaremos ADN, los valores de las características de los individuos. Y como hemos mencionado más arriba como codificaremos las soluciones.

\* Codificación genética de la solución: Es una codificación entera en que cada posición es una característica de la criatura.

\*Operador de cruce: Los hijos heredan características completamente aleatorias de los padres.

OPERADORES DE MUTACIÓN??

\*Critero de parada: Una vez que las criaturas cada generación vuelven a ser las mismas continuamente.

FALTARIA ALGO MÁS??

# Planteamiento Inicial

Realmente, aquí mas que la descripción del modelo que ya habríamos hecho antes, iría la descripción de los distintos parámetros del algoritmo evolutivo, así como la explicación del genoma inicial. Es decir, habría que explicar que hacíamos combate directo entre las criaturas, que hacíamos selección por ruleta, cruce con media aritmética…Además, acuérdate que al principio no teníamos ninguna manera de hacer estadísticas sobre la población.

Inicialmente decidimos empezar con un ADN formado por una lista de 30 números enteros representando cada uno la cualidad que mejoraban, y 20 números en coma flotante, representando las ponderaciones de las acciones a realizar, y comprendidos en el intervalo (-10,10). Al principio parecía la opción más lógica teniendo en cuenta lo que representaba cada número. Además, descartamos la opción de tener cinco números representando la cantidad de cada característica, porque entonces complicaría mucho más las operaciones de cruce (al tener que descartar todo individuo que no tuviera una suma de 30). En realidad, el hecho de descartar individuos generados a partir del cruce no es algo “prohibido” en los algoritmos evolutivos (vendrían a ser en la naturaleza criaturas con defectos genéticos que mueren prematuramente), pero si es cierto que era complicar en sobremanera la operación de cruce y mutación.

Como operador de selección escogimos la ruleta, ya que era el más simple de implementar, y para una primera aproximación sobre la que ir observando resultados era válido. Esto supondría que, como se explica en la introducción de los conceptos de algoritmos genéticos, las criaturas más dominantes en las primeras generaciones arrastraran a toda la población a ese máximo relativo.

Para el cruce escogimos cruce uniforme (seleccionar uno de los parientes) en el caso de los genes relacionados con características físicas, y una media aritmética simple para el caso de los genes que moderaban el comportamiento, lo cual nos llevaría a un problema de convergencia al cero.

La mutación venia dada para cada gen de manera individual (es decir, cada gen tenía una probabilidad de mutar de manera independiente al resto de genes del individuo), dada por un ratio inicialmente en un 3%, pero restringido únicamente a los genes de comportamiento, lo que nos llevaría a otro problema de diversidad con las características físicas. En el caso que se diera el caso de mutar, el gen podía moverse en un rango de (-1.5,1.5) con respecto a su valor original, siempre respetando los rangos de (-10,10).

Nos vamos a centrar en el planteamiento y no en la implementación como tal, es decir, vamos a describir los elementos de lo que está compuesto nuestro planteamiento.

**Respecto a las criaturas en sí:**

Planteamos el problema de la siguiente manera, definimos a las criaturas con un ADN el cual éste, está compuesto por:

* El número de acciones que puede realizar la criatura.
* El número de indicadores que serán aquellas características por las cuales las criaturas harán una u otra acción.
* HP: Los puntos de vida de cada criatura, de forma que si llegan al cero significa que la criatura está clínicamente muerta.
* AT: Los puntos de ataque. Éste es el daño que es capaz de infligir una criatura a otra.
* AR: Los puntos de la armadura de la criatura. Esto se traduce a cuánta defensa tiene la criatura y por ende cuánto daño puede soportar antes de que su HP baje.
* EN: Los puntos de energía de la criatura, de forma que cuantos más puntos tengan, más acciones podrán realizar o podrá realizar la acción que quiere.
* AG: Los puntos de agilidad de cada criatura.
* Nombre de la criatura.
* Pesos de HP, AG, AT, AR, EN: Serán determinantes a la hora del combate con otras criaturas, pues éstas dictaminarán en gran medida qué acción se lleva a cabo y cuál no.

Las criaturas pueden hacer las siguientes acciones:

* Esquivar
* Bloquear
* Atacar
* Pasar

Por otro lado, también tenemos en cuenta el peso de su armadura.

**Respecto a la población:**

Si bien podemos definir holgadamente el número de individuos que la componen, de ésta sacamos unos valores que serán clave para más adelante, como por ejemplo para la cuestión de la diversidad de las criaturas. Por ello, calculamos:

* La “media”: Es decir, el valor medio de las características de la población. ESTO SE PUSO A LA PAR CON LA DESVIACIÓN ESTÁNDAR O FUE ANTES Y DE FORMA INDEDIENTE??
* La desviación estándar (SD): La desviación de cada criatura –sus características- con respecto a la “media”. ESTO FUE DESPUÉS lo quito no?

Ahora bien, dejando aparte los cálculos, también podemos sustituir ALMASCERCANO EXTRAERCRIATURA INSERTARCRIATURA ¿?NO PONEMOS NADA VERDAD?

**Respecto al combate:**

Asignamos a cada criatura su vida inicial y su energía inicial, así como ponemos daño cometido a cero. Después las criaturas lucharán en combates de dos durante 50 rondas y siempre que las dos estén vivas.

El resultado de cada combate puede ser: Victoria, Empate, Derrota.

Cabe mencionar que dentro de la acción de Esquivar, ésta está condicionada por la probabilidad junto con la agilidad de la criatura.

PONEMOS TODO LO QUE HAY DETRÁS DEL COMBATE, ES DECIR, PONER POR ESCRITOS TODOS LOS IF’S DEL COMBATE??

CREO QUE AQUÍ DECIMOS TODO LO QUE NOS FALTABA POR ACLARAR EN MÉTODO PARA ADAPTAR BLABLABLA

# Problemas y soluciones

Los primeros problemas que tuvimos con el planteamiento inicial estaba relacionados con la diversidad de los individuos y con la manera en la que los seleccionábamos y cruzábamos en las generaciones siguientes. El hecho de utilizar ruleta significó que los individuos que obtenían alguna mínima ventaja (lo cual era fácil teniendo en cuenta que al principio todas las puntuaciones eran muy bajas), adquiría una gran porción de la ruleta, y por ende acababa repetido muchas veces en la generación siguiente.

Con el paso de las generaciones, esto provocó un fallo en la función de evaluación (el combate); pues al ser iguales todos los combates acababan en empate, con el mismo daño hecho por uno y por otro. Esto es, todos los individuos terminaban siendo iguales y con el mismo valor en la función de evaluación.

Como primera medida para remediar esto cambiamos el operador de selección de ruleta a torneo, para que los individuos con una puntuación considerablemente mayor no acapararan toda la siguiente generación. Este cambio resultó tal y como se esperaba a corto plazo, pero a la larga (al cabo de unas cuarenta o cincuenta generaciones, a diferencia de con ruleta, en la que la población convergía con tan solo cinco o seis rondas) acababa ocurriendo lo mismo.

En este punto, consideramos dos caminos a seguir: por un lado forzar mediante el análisis estadístico en la población a que las criaturas fueran diferentes (posteriormente me adentraré más en este caso), o implantar lo que nosotros llamaremos “población de prueba” o “población de test”, que será a la que enfrentaremos los individuos en lugar de contra sí mismos.

De cara a poder analizar el efecto que cada una de ellas tenía por separado, decidimos implementar primero la población de prueba, pues nos resultaba más fácil en cuanto a programación ser refiere, y preferíamos poder analizar nuevos resultados cuanto antes.

La población de prueba era una población del mismo tamaño que la población normal (que llamaremos para diferenciarla “de evolución”), y en esta primera aproximación se generaba aleatoriamente al principio de la ejecución del programa, manteniéndose durante todo el proceso. Pese a que el problema de convergencia y empate se eliminó, surgió otro problema causado por la implementación de la población de prueba. Al mantenerse durante todo el proceso de evolución la misma población de prueba, las criaturas de la población de evolución conseguían llegar al máximo de puntuación posible, y a partir de ahí no cambiaban apenas. Dicho de otro modo, se estaba entrenando a las criaturas para combatir contra *esa* población de prueba en concreto, pero no contra *criaturas* en general. Es decir, si cambiabas la población de prueba después de haber conseguido una convergencia las mismas criaturas que ganaban con la puntuación máxima ahora no conseguían puntos.

Para evitar esto, la solución era tan simple como ir sustituyendo las criaturas de la población de prueba por criaturas mejores (obtenidas mediante la población de evolución). Además, y aprovechando la otra rama de mejora que nos habíamos dejado antes (análisis estadístico de la población con respecto a sus características físicas), incorporamos una función distancia entre dos criaturas, que nos decía cuanta diferencia había entre cada una de las características físicas entre las dos criaturas (¡!). Esta función era el primer paso para incorporar funciones de media y desviación estándar con respecto a la media de la población, esta última aportándonos ya si una medida adimensional de cuando de diferente era una criatura respecto a la media.

El proceso ahora consistía en sustituir, después del proceso de creación de la siguiente generación evolutiva, las tres criaturas de la población de prueba más cercanas a las tres mejores criaturas de la población evolutiva. Es decir, obteníamos las tres mejores criaturas de la última generación, y cambiábamos la criatura de la población de prueba que más se asemejara a cada una de ellas por dicha criatura. De esta manera, la población de prueba ahora consistía en un grupo de criaturas diversas (porque sustituíamos únicamente la más próxima) y medianamente poderosas (porque eran las mejores de las generaciones anteriores). El problema de que todas las criaturas obtenían la puntuación máxima había sido corregido con esto.

Sin embargo, no estábamos del todo contentos con la diversidad de la población de evolución, pues, aunque no solía haber más de dos o tres individuos iguales (con las mismas características físicas) de una población de veinte, las diferencias eran de un punto arriba en una cualidad y un punto abajo en otra. Por este motivo, decidimos forzar aún más la diversidad penalizando y bonificando a los individuos según lo parecidos que eran a la media. Su puntuación era multiplicada por la desviación estándar, y de esta manera individuos que fueran muy parecidos a la media, pero sin una puntuación mejor que sus vecinos, acababan teniendo pocas probabilidades de sobrevivir en la siguiente generación.

Tenemos que revisar esto. En plan definir bien las etapas (que ha habido más), y explicar bien todo lo que nos pasaba en cada caso. Básicamente las etapas han sido estas:

1. Ruleta, sin población de prueba: Acababan siendo todos iguales y quedaban empate
2. Torneo, sin población de prueba: Idem, pero tarda mas en converger.
3. Torneo, con población de prueba: Las criaturas se quedan subentrenadas. Básicamente como la población de prueba es medio lela, la población real no es forzada a mejorar. Esto se podía ver porque acababan siempre con el máximo e victorias.
4. Torneo, con población de prueba, actualizando la población de prueba: En este punto fue cuando incorporamos el tema estadisico en la población, lo que nos permitió actualizar la población de prueba con nuevos individuos C. C sustituía al individuo mas cercano a él en la poblaicon de prueba. Las criaturas dejaron de estar “subentrenadas”, y casi nunca llegaban al máximo de victorias.
5. Añadir penalización por SD: Mejoraba la diversidad, ya que penalizaba que varias criaturas fueran iguales.
6. Inclusion de las islas, población de prueba se hacia tomando individuos de todas las islas, y cada x generaciones se intercambiaban algunos individuos de algunas islas: versión definitiva, que amplio la diversidad al tener varios sistemas paralelos que convergían a cosas distintas mezclarse de una manera no demasiado “agresiva”, y evitando la convergencia a máximos locales en las islas. Si se daba una convergencia a máximo mlocal en una de las islas, cuando se cruzaban entre islas el máximo local acaba siendo “absorbido” por los individuos mejores que han “saltado”.

------

Primero hubo problemas con los combates, acaban siempre en empate porque todas las criaturas acababan siendo iguales. Por tanto, no había diversidad. La solución a la que llegamos es pasar del método de selección de la ruleta al torneo. Sin embargo, el problema continuaba siendo el mismo. De modo que probamos a usar el concepto de “población de prueba”. Ésta se basa en hacer una población externa y paralela a la que manejamos, y con la cual vamos sustituyendo unos cuantos individuos.

ME EXTIENDO MÁS??

Una vez introducida la población de prueba, probamos el método de selección de ruleta. Sin embargo continuamos con el mismo problema. De modo que llegamos a la conclusión que la raíz del problema y lo que había que atacar cuanto antes es la falta de diversidad. Para ello introducimos la “desviación estándar” (SD). La desviación de cada criatura –sus características- con respecto a la “media”. (Ni que decir tiene tuvimos que introducir también el cálculo de la media.) De forma que cuánto más diferentes fueran las criaturas a sus semejantes más porcentaje se les fueron asignados. Aunque cabe recalcar que ya de por sí su cálculo representó un desafío. Al principio pensamos en crear un genoma medio mediante las diferencias de las característica con el pudiésemos hacer los cálculos. Pues eran con las diferencias de las características de los genomas con los que estábamos haciendo los cálculos. Después de echarle varias vueltas a Jorge se le ocurrió la brillante idea de hacer la SD mediante la media de las características de la población no con las diferencias de los genomas. Obteniendo así de una forma u otra, una “criatura media” con la que podíamos calcular de forma más eficazmente la SD sin alterar demasiado el código ya escrito. DIME SI ME HE COLADO LOCO.

Con todo ello, conseguimos que salieran criaturas diferentes, sí, pero con diferencias minúsculas. Estas diferencias, con el paso de las generaciones, convergían a cero. Con lo cual no salían las criaturas lo suficientemente diversas como queríamos. Además que tuvimos un problema con una parte del Genoma, el número de victorias??? O ERA OTRA COSA ME BAILA MUCHO (ERA LA ULTIMA O ANTEPENULTIMA COLUMNA DE LA MATRIZ GIGANTESCA). El caso es que se repetía para todos los individuos cuando tenía que ser diferente.

Decidimos probar con el concepto de las islas, y hacer que el tipo de los genomas sean números enteros en vez de reales, para que así no converjan a cero la desviación estándar y se agudicen las diferencias. Respecto a las islas, decidimos poner 10 islas en las que van saltando cada X generaciones 6 individuos; para que así no cayésemos en el vicio de intercambiar demasiados individuos entre las islas y terminase comportándose nuestra simulación como si se tratase sólo de un entorno, esto es, de una sola isla.

POR ÚLTIMO?? Dimos otra vuelta de tuerca y en vez de tener una población de prueba para cada isla, pensamos en una población de prueba común para todas las islas pero cada X generaciones se vuelve a generar de nuevo tomando individuos de cada una de las islas, de esta manera ésta se nutre de individuos de todas las islas.

# Resultados finales

PREGUNTA, DONDE PONEMOS LA LINEA ENTRE INICIAL- MEDIO Y MEDIO,FINAL? HE PUESTO TODO EN EL APARTADO DE ARRIBA Y LUEGO SEPARAMOS Y AÑADIMOS FINAL

Aquí iría directamente a resultado final y análisis. Analizamos todos los gráficos, comentamos todo, y tratamos de explicar lo que se ve en los gráficos.

# Comentarios sobre el Resultado Final

Hasta qué punto se han alcanzado los objetivos planteados:

Qué ha quedado sin conseguir o ha sido modificado:

Qué más se podría hacer si se quisiera continuar:

Estoy de acuerdo. También comentar posibles áreas de mejora y ampliación en el futuro.

# Conclusión