

Introducción a la robótica inteligente

Trabajo obligatorio 2

Vinlorenz Mindaros Sanchez

Daniel Ortega Fernández

Grupo 15

Índice

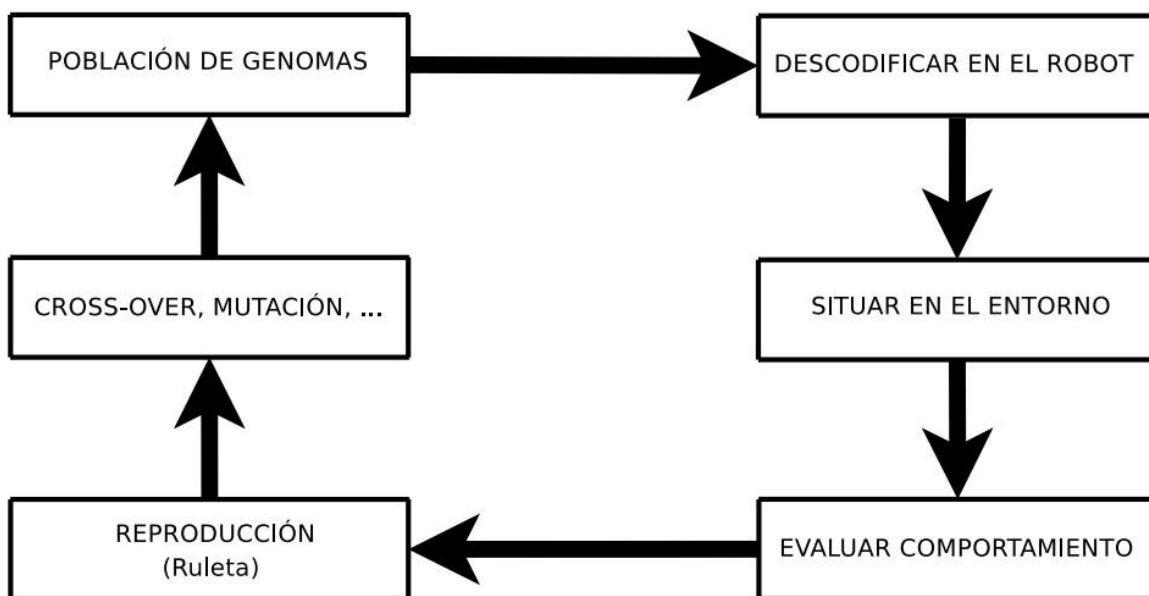
1.- Introducción y descripción del proyecto	3
2.- Primera evolución (fitness 1)	4
3.- Segunda evolución (fitness 2)	5
4.- Tercera evolucion (fitness 3)	6
5.- Cuarta evolución (fitness 4)	8
6.- Quinta evolución (fitness 5)	10
7.- Sexta evolución (fitness 6)	11
8.- Séptima evolución (fitness 7)	12
9.- Octava evolución (fitness 8)	15

1.- Introducción y descripción del proyecto

Bajo la petición de realizar un desarrollo de robótica evolutiva, en nuestro grupo hemos trabajado en dicha tecnología creando una serie de programas guiados por una fitness, la cual indicará cual de nuestros robots será más apto.

La evolución funciona de la siguiente manera, en primer lugar se crea un entorno en el que estarán los robots, y se le dotará a nuestro programa de un algoritmo genético con su respectiva red neuronal. Basándose en esto nuestro programa generará una población de robots aleatoriamente otorgando valores a los cromosomas de nuestros robots individualmente. Hecho esto se evaluarán estos cromosomas según la fitness anteriormente mencionada la cual no es más que una función matemática. Una vez evaluados se cogerán los mejores individuos y pasarán a la siguiente generación sin modificaciones, mientras que los demás individuos serán mezclados entre sí para generar individuos nuevos en la generación siguiente (hijos).

Este proceso se repetirá hasta el objetivo de generaciones descrito en dicho programa.



2.- Primera evolución (fitness 1)

Como primera evolución hemos partido de un sistema básico en el cual el robot únicamente pasará los valores obtenidos por sus sensores de proximidad y de luz amarilla.

La fórmula empleada para diagnosticar lo bueno que será un individuo de la generación será la siguiente:

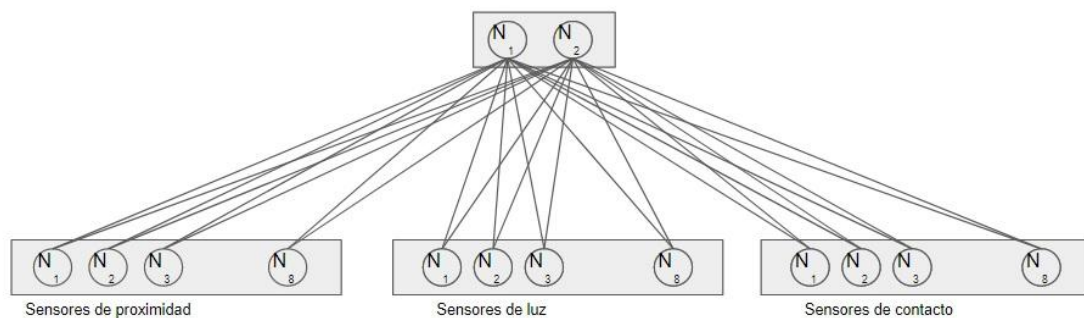
$$F1 = \frac{\sum_{i=0}^{Nsteps} (L0 \times L7 \times (1 - \max\{PROX\}) \times (1 - \max\{CON\}) + \max\{L\})}{Nsteps}$$

Para la creación de la población hemos tomado los siguientes valores:

- Número de la población: 100
- Número de generaciones: 200
- Valor de mutación: 0.1
- Número de élites: 10

Hemos tomado estos valores muy aproximados ya que al tratarse de la primera prueba no teníamos ningún antecedente de qué valores serían los apropiados.

Teniendo en cuenta los sensores que vamos a evaluar en nuestra función de fitness nos quedaría el siguiente diagrama:



Con todo esto procedemos a evolucionar el robot, obteniendo un robot que básicamente no realiza ninguna función, solamente se mueve por el mapa un poco en dirección a la luz marcha atrás.

Esto pensamos que se debe a la declaración de la fitness por lo cual vamos a desarrollar otra diferente.

3.- Segunda evolución (fitness 2)

En esta evolución nos hemos enfocado en lograr que el robot no se moviese marcha atrás, para ello hemos incluido en la fitness un valor que evalúe la velocidad de ambas ruedas y así lograr que cuando el robot no se mueve o se mueve marcha atrás obtenga menos valor de fitness y sea evaluado como peor.

Para ello la nueva función de fitness será:

$$F2 = \frac{\sum_{i=0}^{Nsteps} (L0+L7+(1-\max\{PROX\})+(1-\max\{CON\})+\max\{L\})\times\frac{1}{5}V}{Nsteps}$$

Todos los demás valores serán idénticos a la anterior evaluación.

Como resultados de esta evolución hemos obtenido un robot muy parecido al anterior pero que se mueve ya para adelante y parece que se mueve en dirección a la luz.

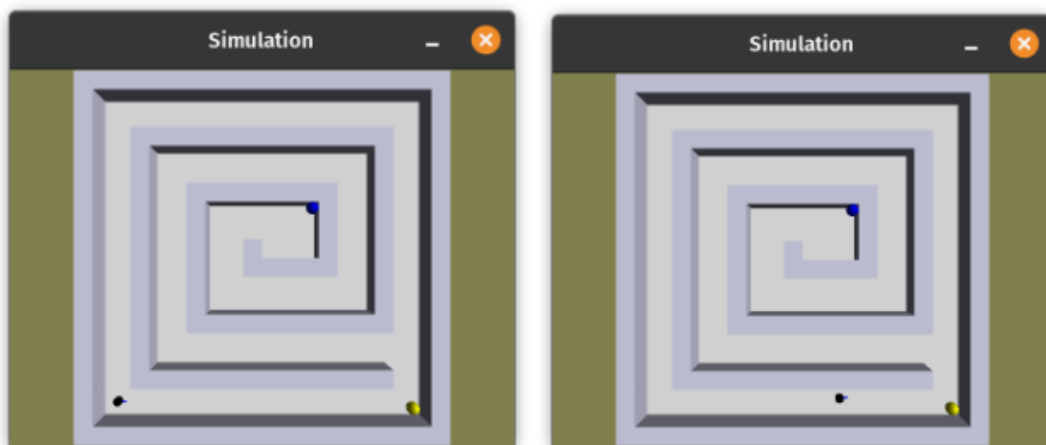
4.- Tercera evolución (fitness 3)

En esta fitness usamos los mismos sensores que en la anterior pero cambiamos un poco la función y evolucionamos al robot en un mapa con forma de espiral. Para ello empleamos esta ecuación:

$$F3 = \frac{\sum_{i=0}^{Nsteps} \frac{1}{4}((1-\max\{PROX\}) \times (1-\max\{CON\}) + \max\{L\} + L0 + L7)}{Nsteps}$$

Como podemos observar es algo diferente a la anterior, puesto que hemos decidido quitar la parte del movimiento de las ruedas y enfocarnos en darle unos pesos un poco diferentes a los sensores para así, junto con la evolución en un mapa en espiral poder obtener un mejor resultado.

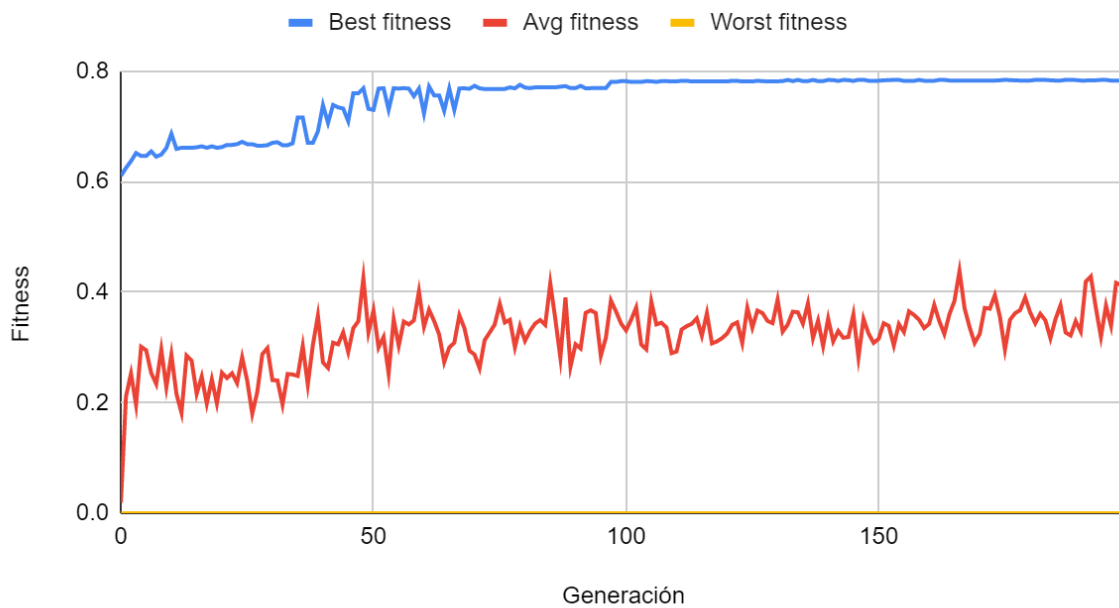
Los resultados de dicho robot han sido los siguientes:



El robot se mueve adecuadamente hacia la luz pero llegado a un punto pendula hacia adelante y atrás cuando situamos la luz en el centro de la espiral, sin llegar nunca al centro de la espira.

Debido a que este robot ya realiza algunas tareas meridianamente decentes hemos decidido comenzar a analizar también los datos de cómo evolucionan los individuos de cada generación, los cuales podemos observar en la siguiente gráfica:

F3 - Mapa espiral



Como podemos observar desde un inicio ya tenemos un valor de fitness muy alto lo cual normalmente no es buena señal, también se aprecia que obtiene valores casi máximos muy pronto, por tanto podemos seguir concluyendo que el problema, a priori, de nuestra evolución se trata de la función de fitness de nuevo.

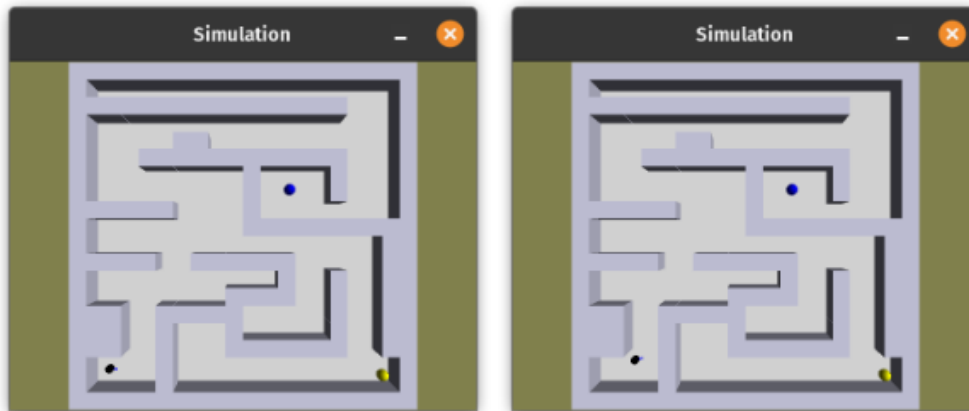
5.- Cuarta evolución (fitness 4)

En esta evolución hemos optado por evolucionar al robot en un entorno laberíntico y hemos cambiado la fitness a la siguiente:

$$F4 = \frac{\sum_{i=0}^{Nsteps} \frac{1}{3}((1-\max\{PROX\}) \times (1-\max\{CON\}) \times (\max\{L\} + L0 + L7))}{Nsteps}$$

Estos cambios están realizados para otorgarle un mayor impacto a los sensores de luz y así que el robot responda mejor ante este estímulo. Por otro lado, el uso del laberinto procura evitar el movimiento que el robot realizaba en la anterior prueba.

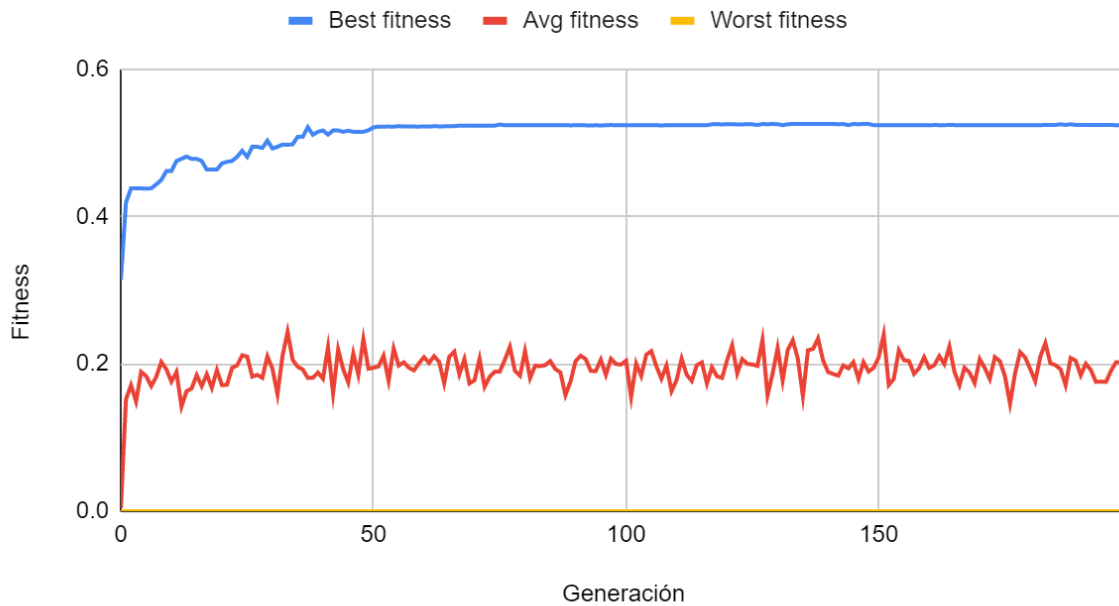
Los resultados del mejor robot de esta evolución han sido los siguientes:



El robot se mueve con buena dirección hasta el lugar de la captura de la derecha y ahí se queda totalmente parado. Intuimos que esto se debe a la fitness y a no dar suficiente tiempo de evaluación al robot. Se podría solucionar fácilmente añadiendo una condición de movimiento del robot como hemos hecho anteriormente.

Por otro lado, al igual que en la prueba anterior hemos analizado las evoluciones de los robots de las generaciones obteniendo esta tabla con dichos resultados:

F4 - Mapa laberinto

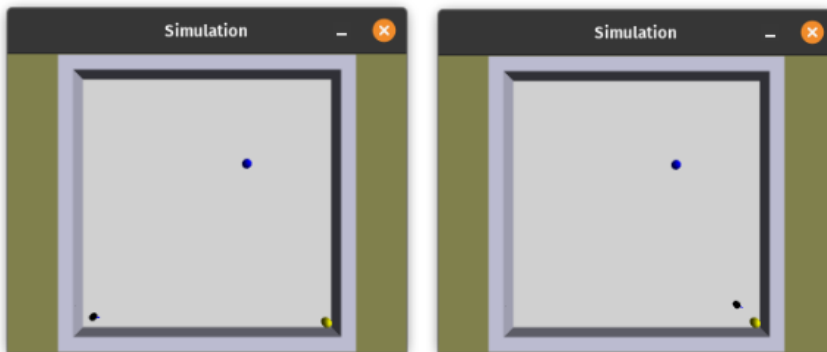


Como podemos observar en esta evolución la generación inicial empieza con menos valor de fitness lo cual a primera vista es bastante positivo, no obstante se obtiene un valor bastante elevado de la fitness bastante rápido lo que apunta que nuestra fitness podría ser poco exigente con el robot y por eso se consiguen resultados muy positivos muy rápido en cuanto a generaciones se refiere.

6.- Quinta evolución (fitness 5)

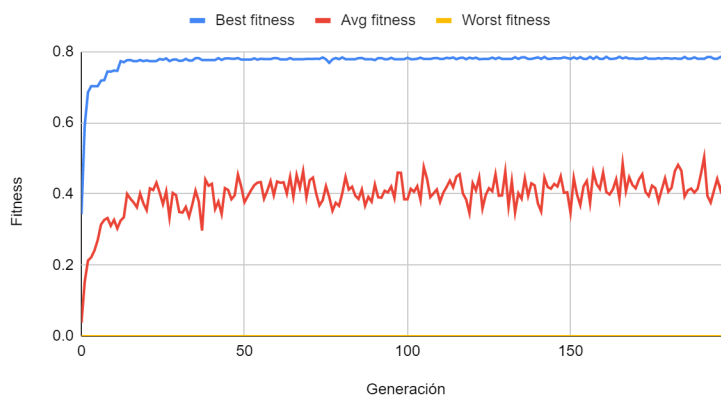
En esta fitness intentamos centrarnos exclusivamente en que el robot se acerque a la luz amarilla, esto lo realizamos para comprobar que nuestra función que provoca que persiga la luz amarilla es correcta.

$$F5 = \frac{\sum_{i=0}^{Nsteps} \frac{1}{3}((1 - \max\{PROX\}) \times (\max\{L\} + L0 + L7))}{Nsteps}$$



En esta simulación el robot se acerca hasta la luz para pararse a la distancia de la segunda captura, esto se debe a la función de fitness que, al igual que en el anterior ejemplo al no tener una condición de movimiento de las ruedas puede pararse al lado de la luz y así obtener mucho valor de fitness.

F5 - Mapa vacío



Aquí podemos ver como la evolución empieza con valores muy rápidos de fitness los cuales aumentan muy rápido en el tiempo indicando al igual que en las anteriores que nuestra fitness es muy poco exigente y de esta manera no deja que nuestro robot siga evolucionando adecuadamente, ya que obtiene un valor muy alto en seguida y luego no cambia prácticamente nada el comportamiento de nuestros robots.

7.- Sexta evolución (fitness 6)

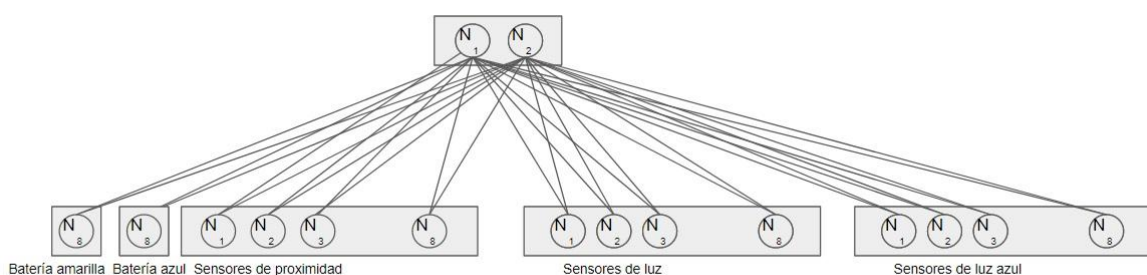
En esta evolución hemos decidido añadirle otra luz de color azul para provocar que el robot se mueva de una luz a otra, al principio nuestras ecuaciones de fitness provocaban que, o bien el robot iba hacia la luz más cercana o bien el robot se quedaba parado a una distancia intermedia de las luces, por tanto pensamos en dos soluciones posibles, una era hacer que el robot no se quedase parado al igual que hicimos en una de las fitness anteriores, pero habia la opción de que se quedase dando vueltas. Por tanto decidimos probar con la otra solución que se nos ocurrió, y esta es incorporar ahora las baterías del robot, tanto la azul como la amarilla. La intención de esto era provocar que cuando una batería se consumiese el robot fuese hasta la batería de dicho color para recargarse. Esto lo hemos intentado con esta función de fitness:

$$F6 = \frac{\sum_{i=0}^{Nsteps} coef1 \times (1 - \sqrt{\Delta v}) \times ((\max\{PROX\}) + coef2 \times battery + coef3 \times Bbattery + coef4 \times \max\{L\})}{Nsteps}$$

Como se puede ver empleamos varios coeficientes, esto lo usamos para darle un mayor o menor peso a los diferentes sensores del robot.

Al modificar esto hemos cambiado el número de los cromosomas por usar los sensores de batería y aumenta su tamaño a 54 si usamos todos los sensores de proximidad y 44 si noh (hemos probado ambas soluciones y la mejor ha sido no usarlo).

La arquitectura neuronal del sistema evolutivo es la siguiente (para todos los sensores de proximidad):



Con todo esto el comportamiento de nuestro robot ha sido básicamente desastroso, seguramente provocado por los coeficientes mencionados anteriormente. Cuando los variamos provocamos que el robot se chocase contra muros, se moviese libremente sin hacer caso ni a las luces ni a la batería u otros diversos comportamientos que distaban mucho del esperado.

8.- Séptima evolución (fitness 7)

En esta evolución hemos procurado arreglar los fallos de la anterior partiendo de la misma base, usar las baterías para indicar al robot que luz ha de perseguir.

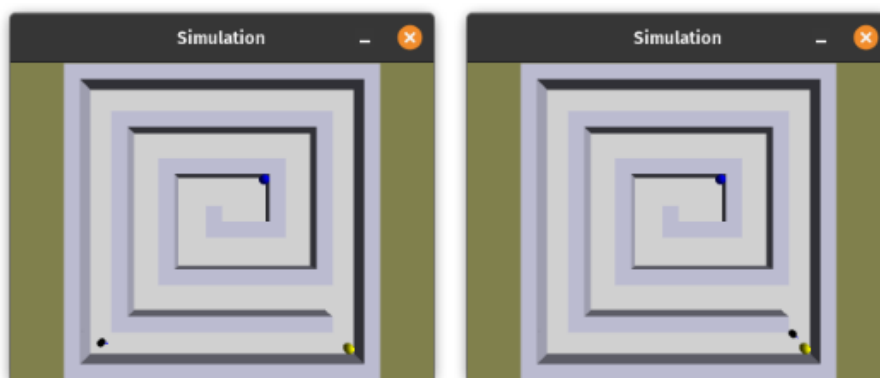
Esto lo hemos logrado con la siguiente función de fitness:

$$F7 = \frac{\sum_{i=0}^{Nsteps} V \times (1 - \max\{PROX\}) \times battery \times \frac{1}{3} (\max\{L\} + L0 + L7) \times Bbattery \times \frac{1}{3} (\max\{BL\} + BL0 + BL7)}{Nsteps}$$

Como se puede apreciar en la fitness hemos dejado a un lado los experimentos con coeficientes debido a los malos resultados de la anterior prueba y hemos decidido incorporar de nuevo las velocidades de las ruedas para así lograr que el robot esté siempre en movimiento en cuanto a la incorporación de las baterías la hemos añadido de tal manera que si una batería llega a acabarse la puntuación de esa luz sería mayor que la de las otras luces y así el robot priorizaría esa luz en dicho momento, esto sumado a la ya mencionada condición de movimiento de las ruedas y la condición de siempre del sensor de proximidad nos ha permitido tener un resultado bastante decente de nuestro mejor robot.

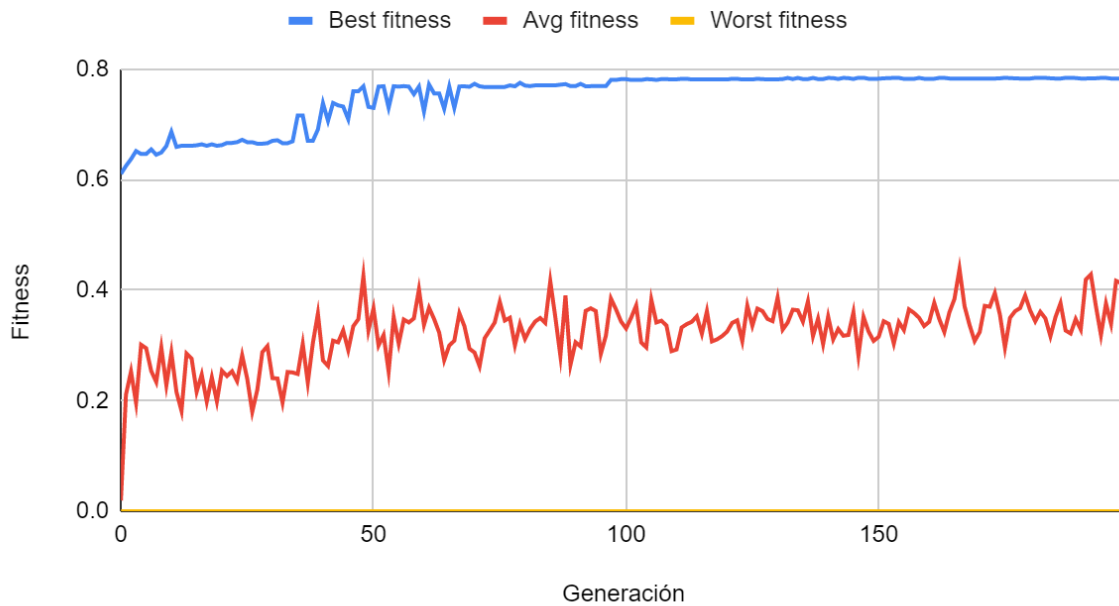
Debido al buen resultado de dicha fitness hemos decidido evolucionarlo con los mapas anteriormente citados en espiral y laberinto.

En cuanto al mapa en espiral obtenemos la siguiente evolución:

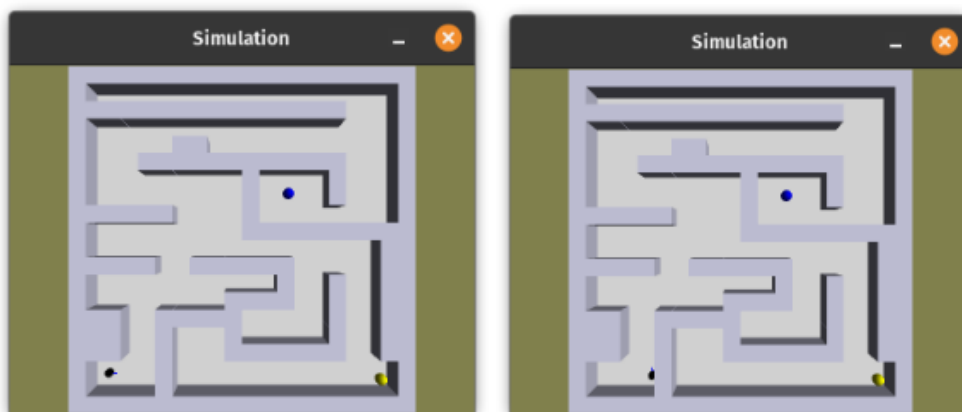


Este ejemplo es muy similar al anterior de espiral, solo que aquí retrocede un poco cuando llega a la luz amarilla debido a la batería de la luz azul pero después se queda completamente parado. No hemos sido capaces de encontrar una causa para que esto pase.

F7 - Mapa espiral

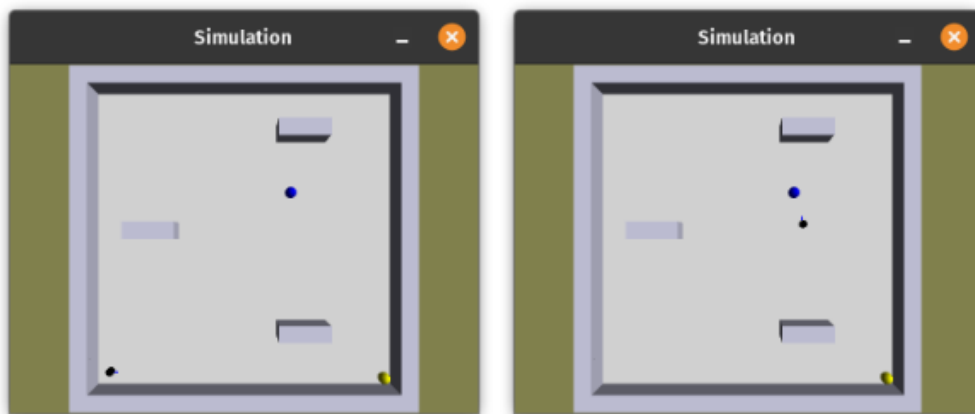
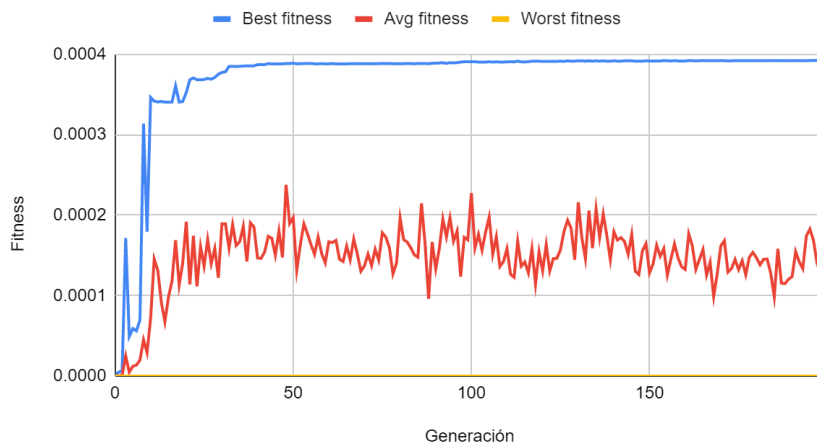


Por contraparte con el laberinto obtenemos la siguiente gráfica:



Similar al anterior, se dirige hacia la luz amarilla, da un movimiento hacia la azul y se detiene, no sabemos si se debe a una mala incorporación de las baterías o a una falta de tiempo de evaluación. Por ello hemos decidido evolucionarlo en otro entorno con menos muros para así ver cómo se comporta.

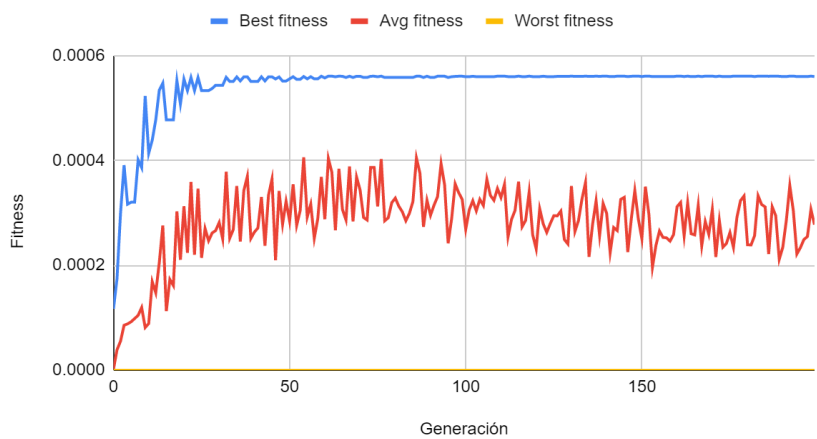
F7 - Mapa laberinto



En este entorno el robot funciona bastante mejor, se dispone hacia la luz azul, se queda dando vueltas y después da un respingo hacia la luz amarilla para acabar deteniéndose.

Su evolución es la siguiente:

F7 - Mapa pocos muros

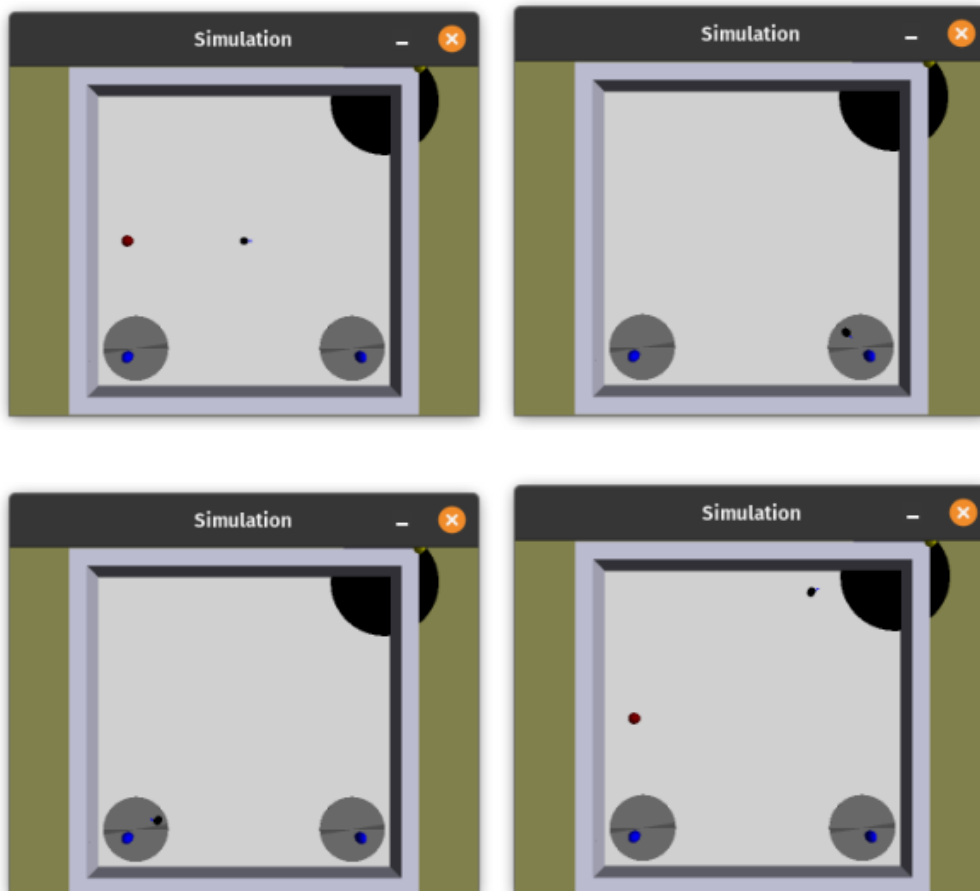


9.- Octava evolución (fitness 8)

Con esta última fitness hemos intentado que el robot realizara una tarea de recolección al igual que en la entrega obligatoria 1. Para ello hemos utilizado sensores de suelo adicionalmente a los ya utilizados anteriormente. Dicha función queda de la siguiente manera:

$$F8 = \frac{\sum_{i=0}^{Nsteps} RB0 \times (\frac{1}{2}(RL0+RL7)) + RB1 \times (GF1 \times (\frac{1}{2}(BL0+BL7)) + GF0 \times (\frac{1}{2}(L0+L7)))}{Nsteps} \times 1 - \min\{\frac{Ncoll,30}{30}\}$$

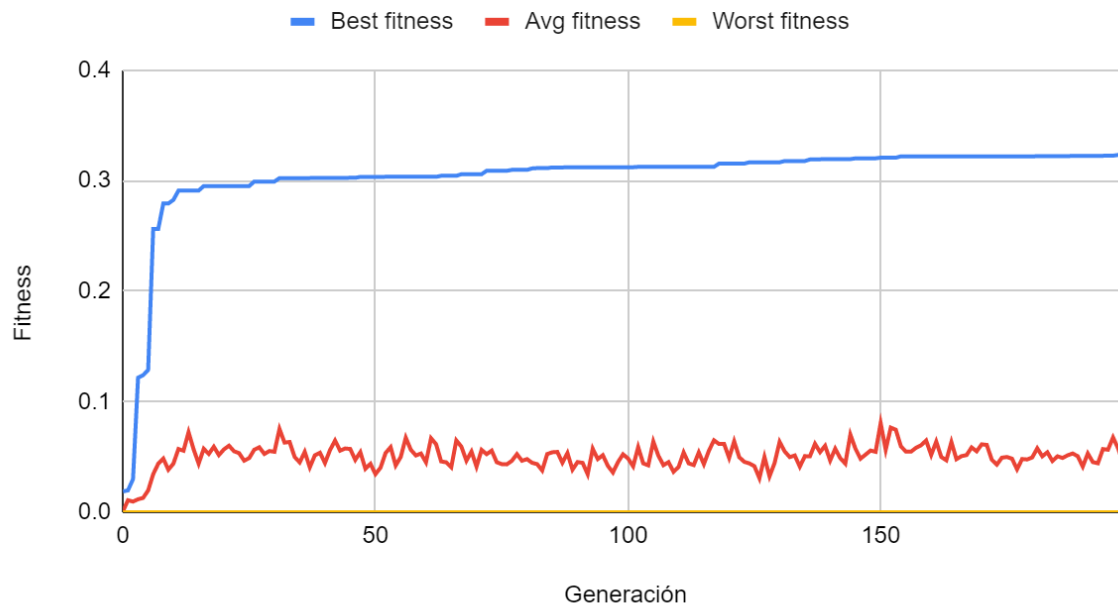
Esta fitness la hemos realizado basándonos en el principio de la anterior de las baterías pero mezclándolo con uno de los ejemplos de clase, en este caso utilizaremos un sensor de suelo, funciona de la siguiente manera, mientras tenga batería por encima de un umbral irá hacia la luz azul hasta encontrar una zona gris, hecho esto cambiará el valor de suelo gris añadiendo uno y activando el flag. Una vez se activa el flag solo sumará fitness ir hacia la luz amarilla, por tanto irá hasta dicha luz y reseteará el flag.



Para esta simulación restaría comprobar si el robot de verdad sigue las luces o únicamente se ha aprendido el mapa. Esto se realizaría cambiando de sitio las luces y las zonas grises o eliminándolas.

En cuanto a los datos de la evolución son los siguientes:

F8 - Mapa vacío



10.- Bibliografía

- *Fundamentos Teóricos de la Asignatura y Manual del Simulador IRSIM*, Apuntes, Robolabo
- *Simulador IRSIM*, Presentaciones de clase, Robolabo
- *Redes neuronales*, Presentaciones de clase, Robolabo
- *Algoritmos genéticos*, Presentaciones de clase, Robolabo
- *Robótica evolutiva*, Presentaciones de clase, Robolabo