Programación Evolutiva
Práctica 1

Miguel Ángel Castillo Moreno

Juan Pedro Olmos Rojas

Descripción del algoritmo

Nuestro algoritmo incluye las siguientes funciones:

- **Representación binaria y con número reales**. Estas están disponibles para cualquier función a optimizar y para cada algoritmo de selección, mutación, cruce, con la excepción del cruce aritmético para la representación binaria.
- **Selección**:
  - Ruleta.
  - Torneo determinístico: con 3 contendientes.
  - Torneo probabilístico: con 3 contendientes y un 70% fijo de probabilidades de escoger al ganador, y un 30% al perdedor.
  - Universal estocástica.
  - Restos: con una k = número de individuos de la población * 1.05. El resto de individuos se seleccionan con el torneo probabilístico.
  - Sin selección: la población se mantiene en lugar de seleccionar individuos.
- **Cruce**:
  - Monopunto: Tanto para bits como reales.
  - Uniforme: Tanto para bits como reales.
  - Aritmética: Solo para reales, con la posibilidad de seleccionar el valor de alpha.
- **Mutación**:
  - Básica para ambos tipos de representación.
- **Elitismo**
- **Estancamiento**:
  - Podemos elegir que, cuando hayan pasado n generaciones sin haber logrado mejorar el mejor individuo, se resetee un k% de la población, para introducir nueva aleatoriedad.

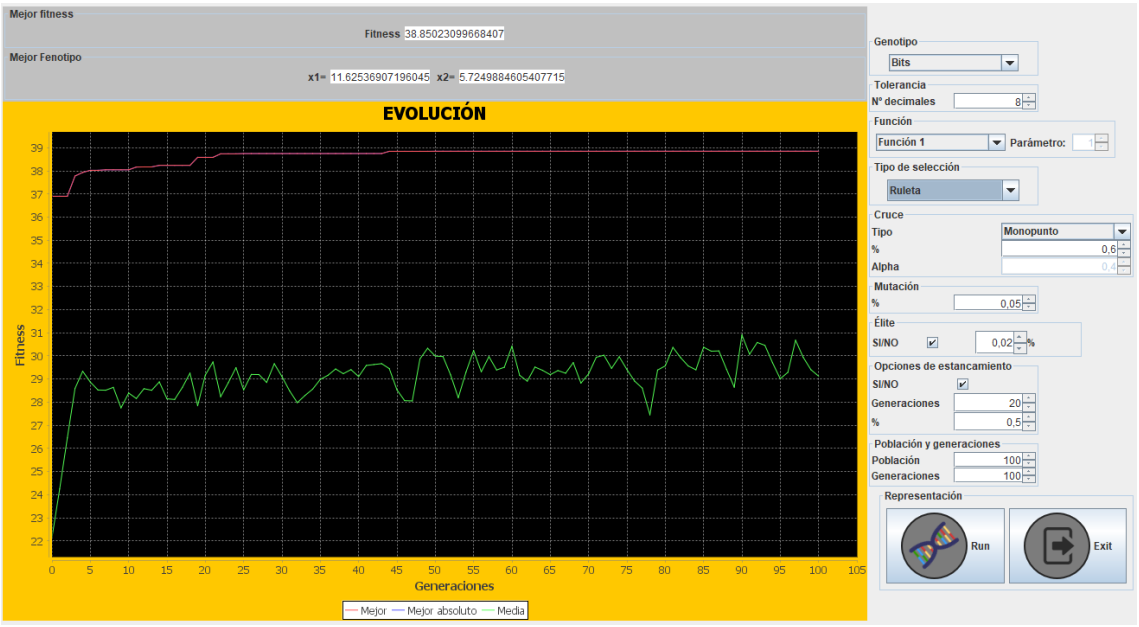
Resultados para cada función

Vamos a mostrar los resultados obtenidos para cada función. Los parámetros elegidos se pueden ver a la derecha de las imágenes, y han sido escogidos para mostrar los mejores resultados obtenibles, intentando ir variando en estos mientras que los resultados sigan siendo buenos. Por ejemplo, mantenemos siempre activado cierto porcentaje de elitismo porque ayuda a obtener muchos mejores resultados, pero vamos cambiando el método de selección.

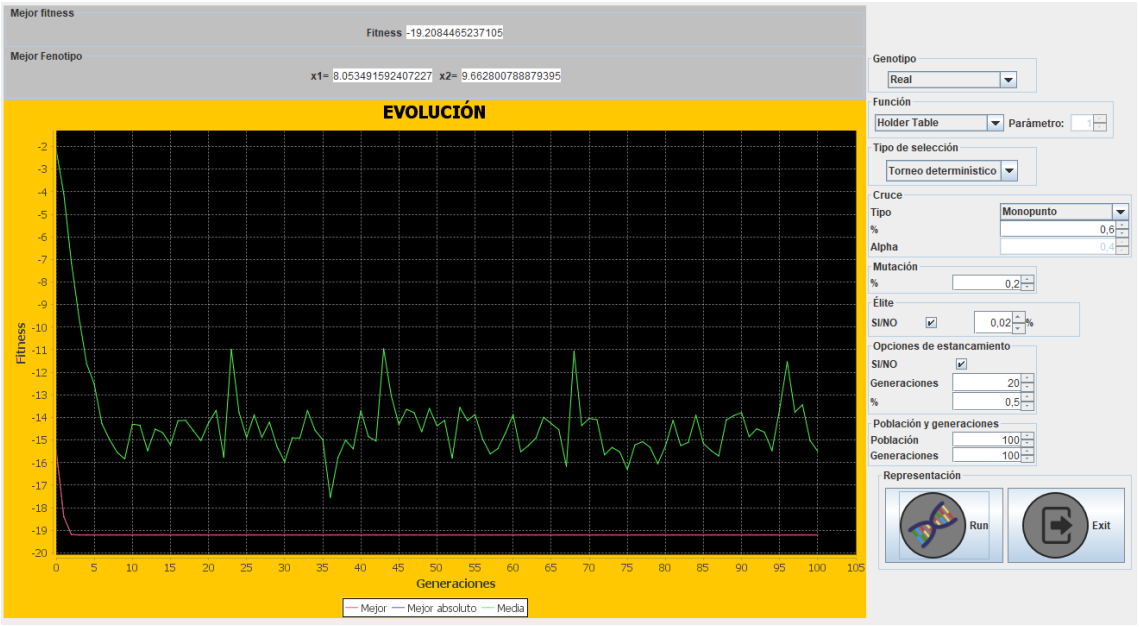
En la parte superior también se muestra el fenotipo del mejor individuo obtenido, y el máximo (o mínimo) fitness para este.

Como podremos observar, la mayoría de estas gráficas llegan a su mejor valor muy rápidamente. En algunos casos, solo con la aleatoriedad de generar la población inicial ya se obtienen individuos muy buenos.

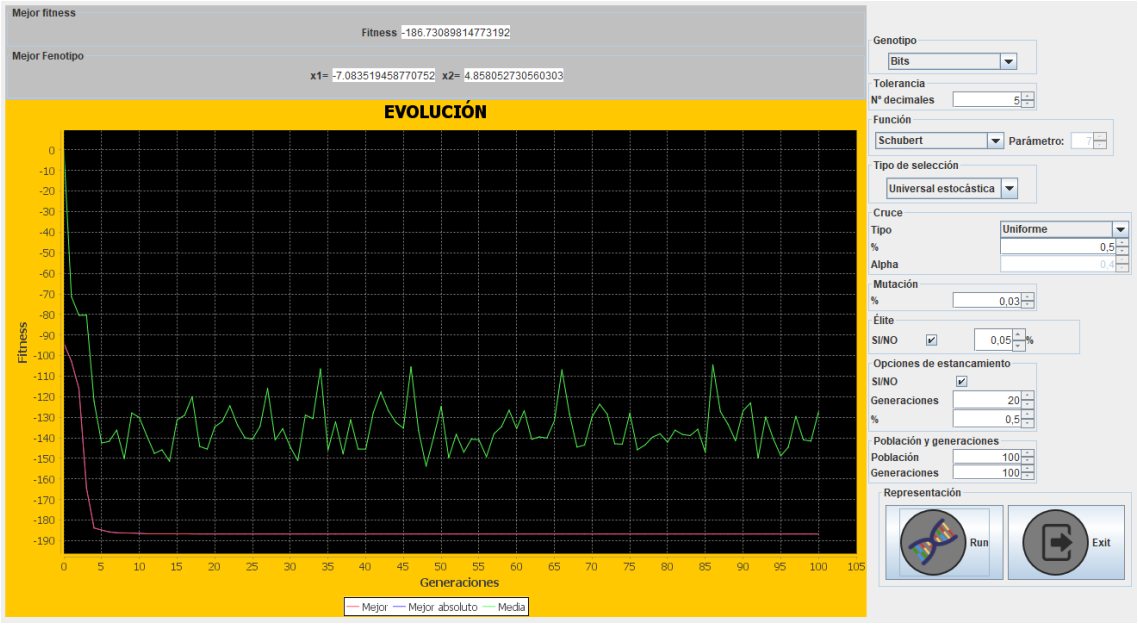
Función 1:



Función 2: Holder Table

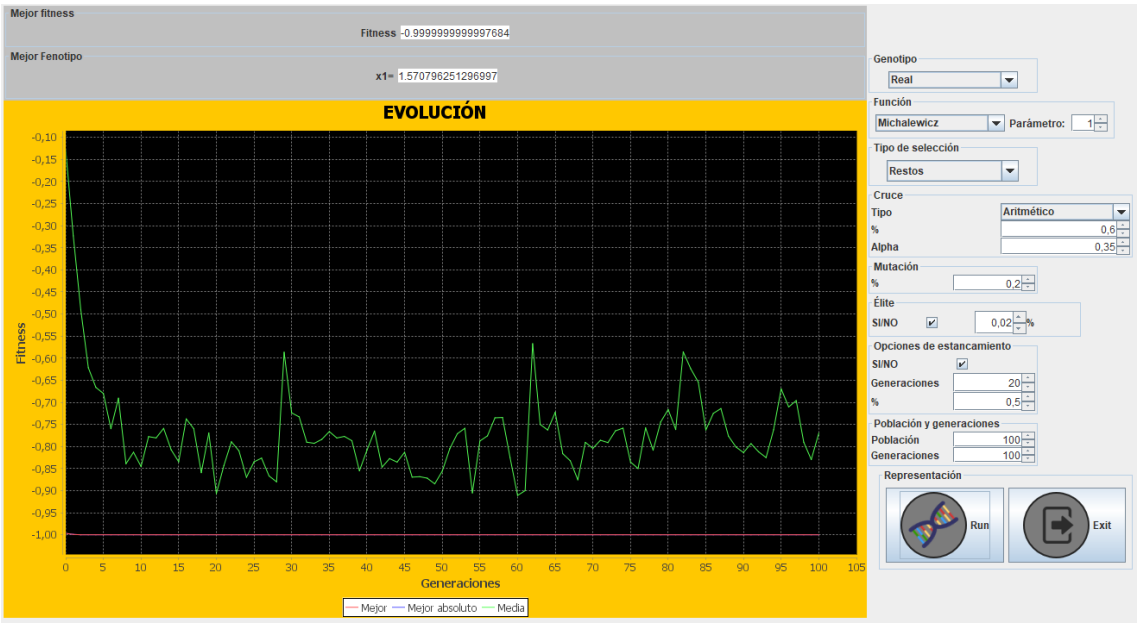


Función 3: Schubert

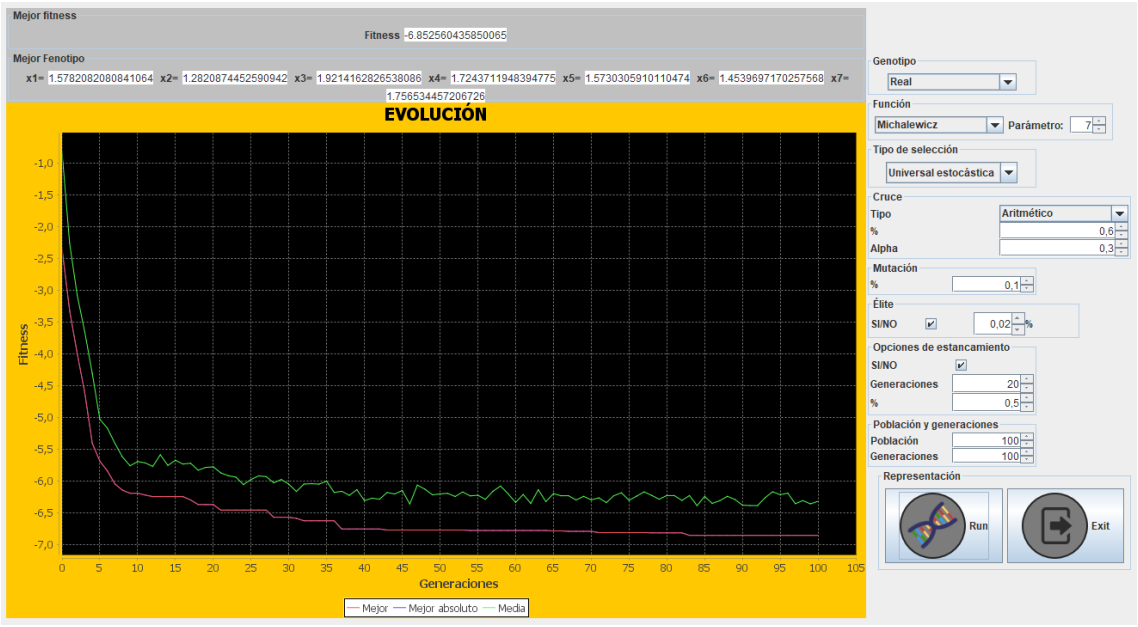


Función 4 Michalewicz

N=1



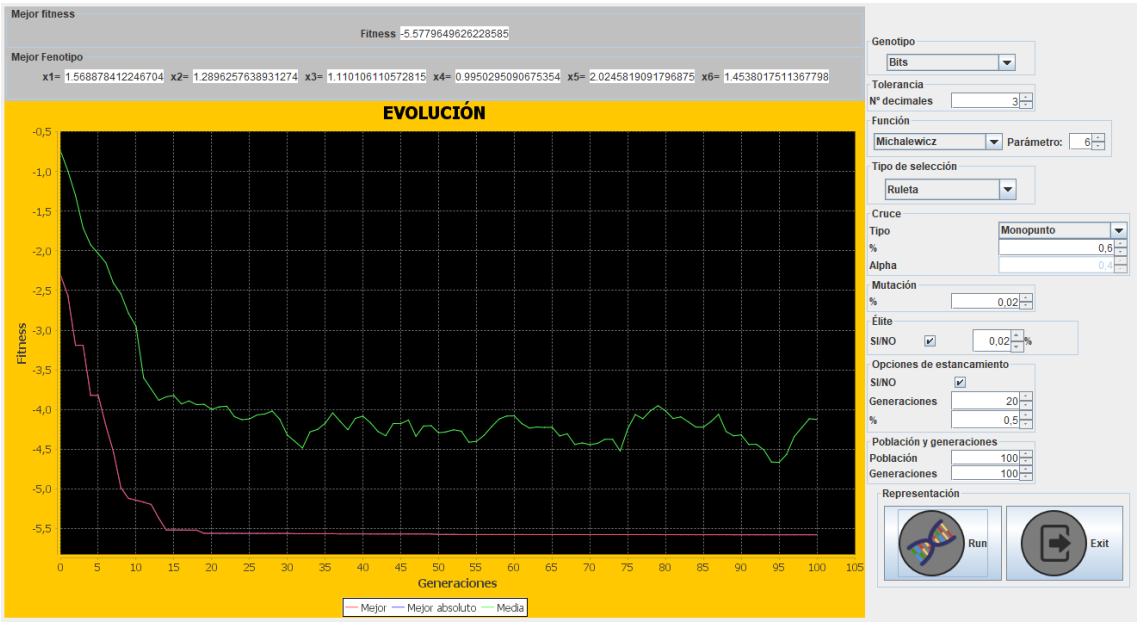
N=7

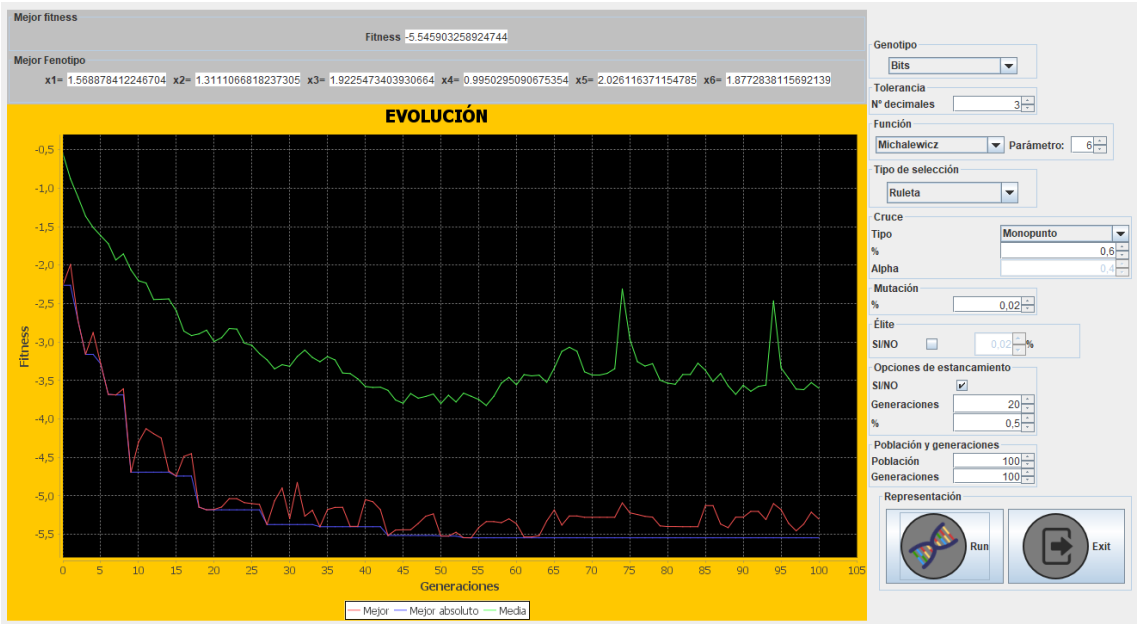

Mejor fitness
Fitness -6.852560435850065
Mejor Fenotipo
x1= 1.5782082080841064 x2= 1.2820874452590942 x3= 1.9214162826538086 x4= 1.7243711948394775 x5= 1.5730305910110474 x6= 1.4539697170257568 x7= 1.756534457206726
EVOLUCIÓN
Fitness
Generaciones
Mejor
Mejor absoluto
Media
Genotipo
Real
Función
Michalewicz
Parámetro: 7
Tipo de selección
Universal estocástica
Cruce
Tipo
Aritmético
%
0,6
Alpha
0,3
Mutación
%
0,1
Élite
SI/NO
0,02
%
Opciones de estancamiento
SI/NO
Generaciones
20
%
0,5
Población y generaciones
Población
100
Generaciones
100
Representación
Run
Exit

Algunas observaciones

A continuación, vamos a analizar algunas de las gráficas obtenidas para observar los resultados. Además, iremos cambiando los parámetros para mostrar diferentes tipos de selección, cruce, representación…

Con y sin elitismo



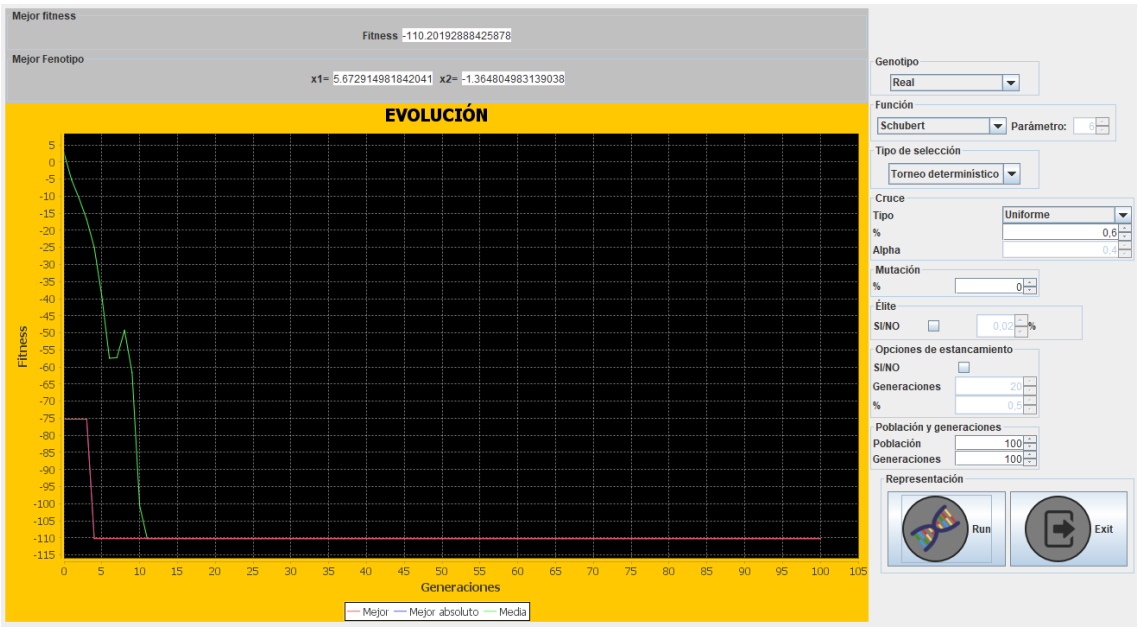


En la gráfica superior, un 2% de los 100 individuos se añaden a la elite, asegurándose así de que pasan a la siguiente generación. En la segunda gráfica, esto no se hace.

Como podemos observar, cuando hay elitismo, el mejor de todas las generaciones y de cada una de ellas siempre es el mismo. Esto se debe a que el mejor va a pasar directamente, por lo que nos aseguramos de quedarnos con él. Cuando no hay elitismo, esto no pasa, y el mejor suele empeorar.

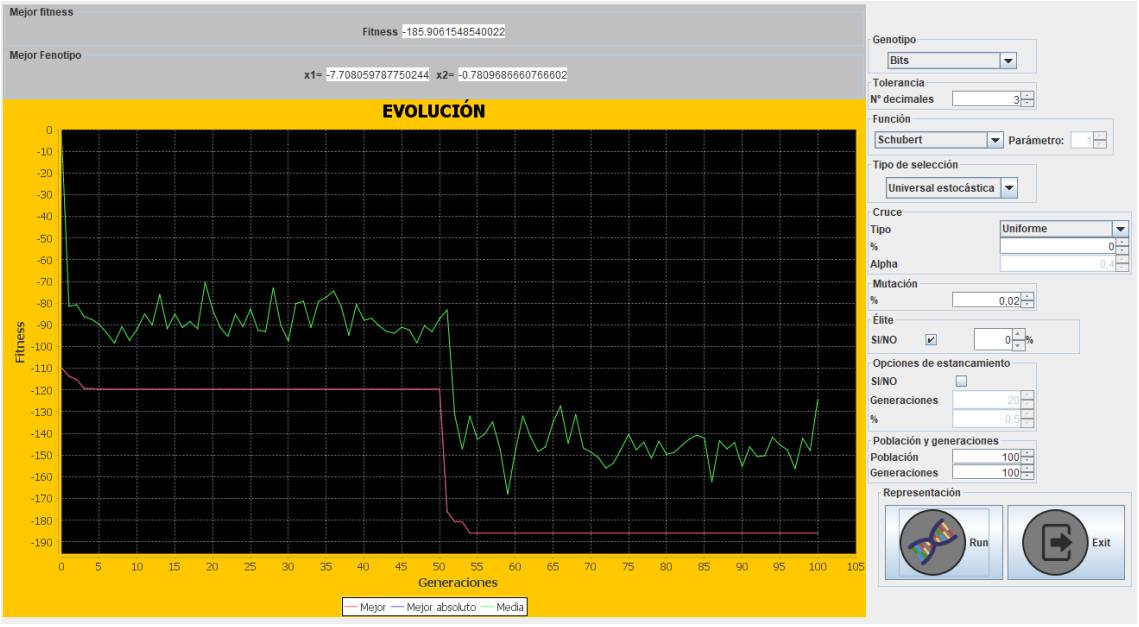
Se puede ver también como el elitismo acelera mucho la obtención del resultado óptimo.

Sin mutación



En este caso hemos desactivado la mutación, dejando solo el cruce y la selección. Se ve como los individuos mejoran, pero sin introducir algo de aleatoriedad, se estanca rápidamente el algoritmo, y al final todos acaban siendo iguales.

Sin cruce

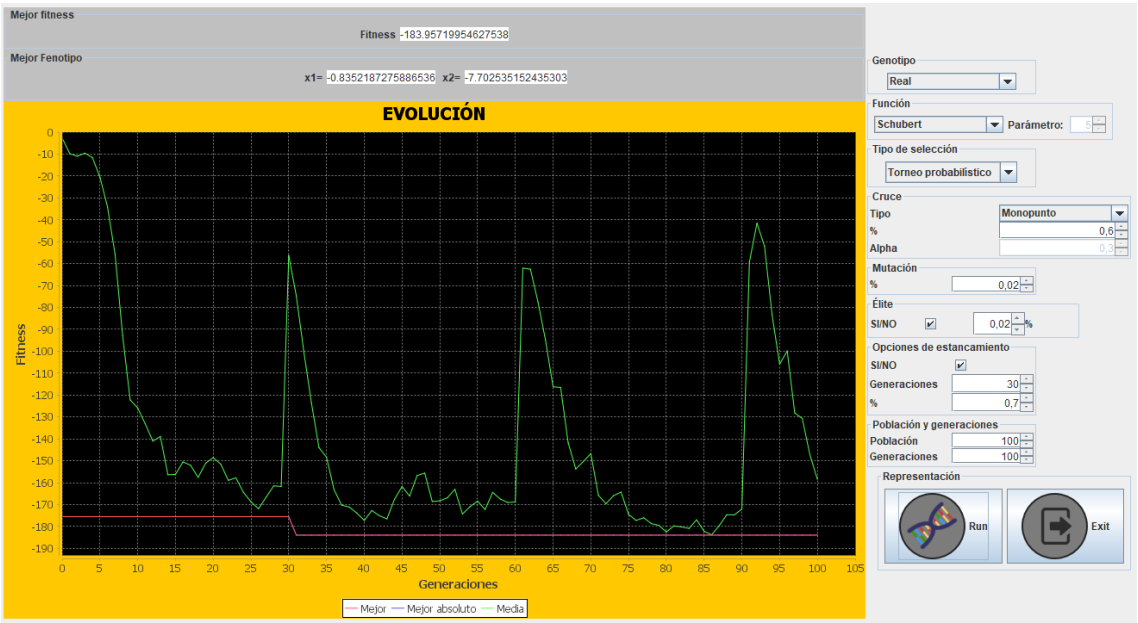


En el caso de que no haya cruce, es muy complicado que el algoritmo vaya mejorando. Se observa como mejora inicialmente gracias a la selección, pero una vez se estanca,

tenemos que tener suerte para que una mutación consiga un individuo mejor. Esto pasa en la generación 51, en la que se mejora drásticamente.

Sin cruces, la mejora es mucho más lenta y complicada, teniendo que confiar más en mutaciones positivas.

Con reseteo por estancamiento

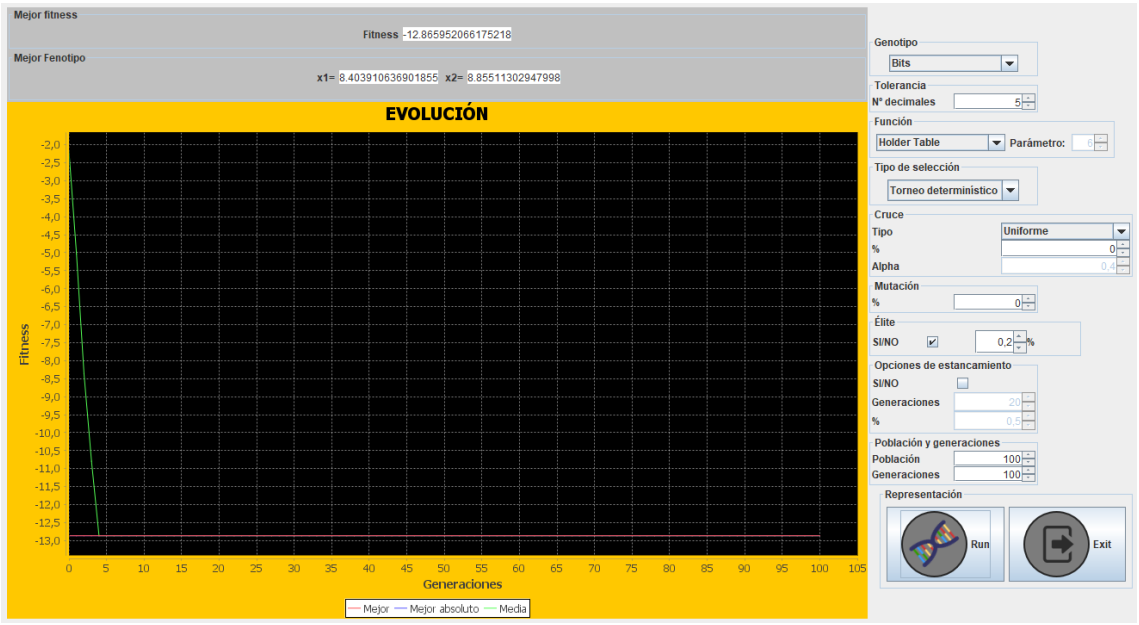


Para la generación de esta gráfica, tenemos todo activado. Casualmente, en la generación inicial había un individuo bastante bueno. Sin embargo, al haber una probabilidad de mutación del 2% en una población real, es muy difícil que haya mutaciones, por lo que la población se estanca.

En nuestro algoritmo hemos añadido una funcionalidad. Con ella, si pasan en este caso 30 generaciones sin una mejora del mejor individuo absoluto, un 70% de la población se resetea, aleatorizando su genotipo.

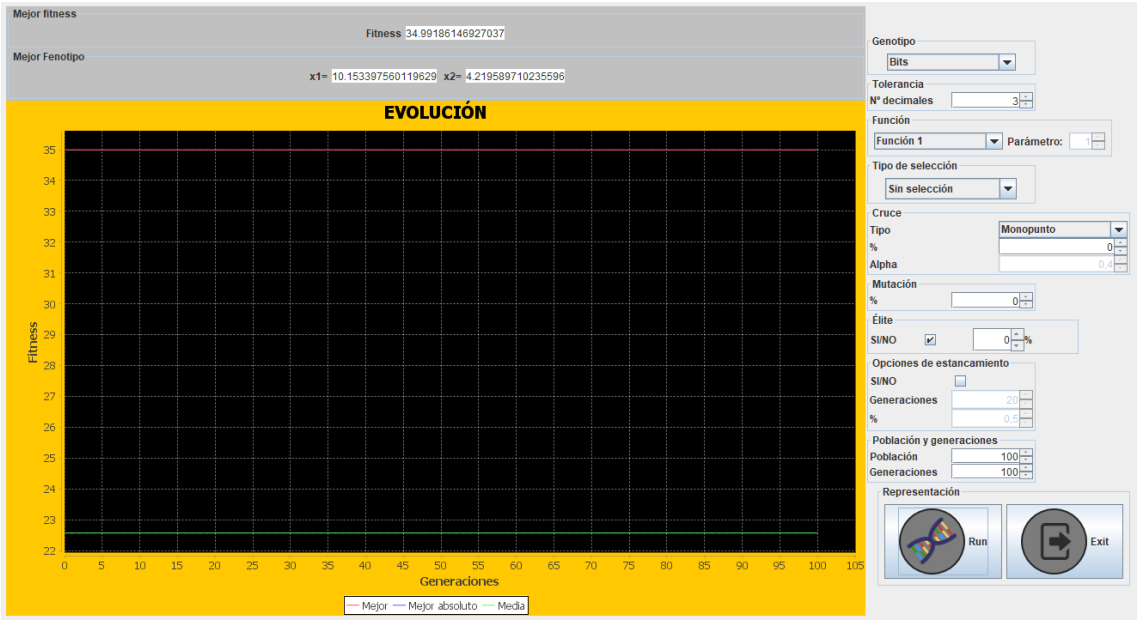
Esta funcionalidad explica las grandes subidas (lo que implica empeoramiento) de las medias. Por casualidad, en el primer reseteo producido en la generación 30, se obtiene un individuo mejor, con los que se obtiene una pequeña mejora.

Sin mutación ni cruce



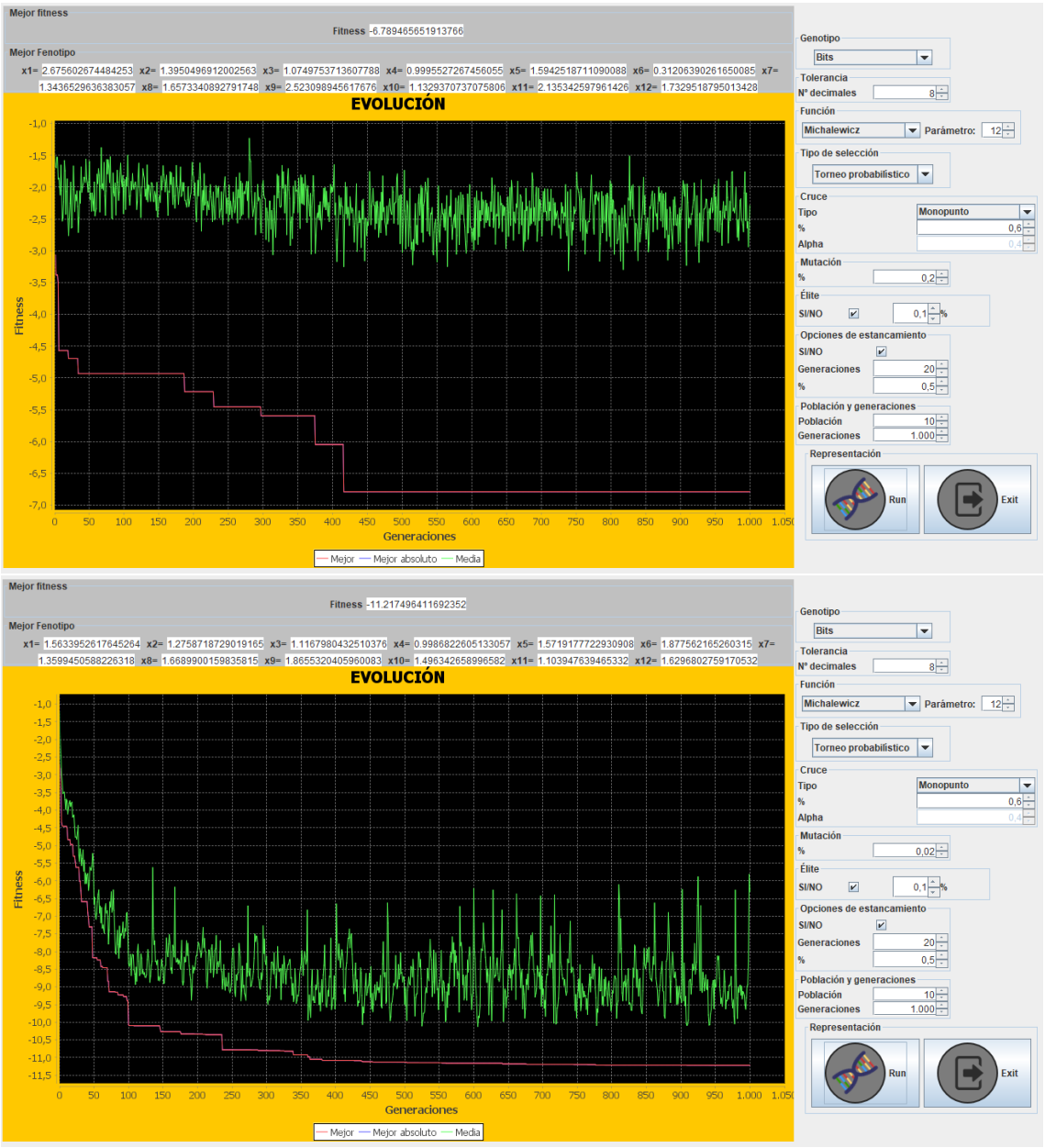
La población va mejorando gracias a la selección al comenzar. Sin embargo, como no hay forma de alterar los individuos, enseguida se obtiene una población con todos los individuos iguales, inhabilitando la posibilidad de mejora.

Sin nada



Por curiosidad, hemos deshabilitado la evitación del estancamiento, el cruce, la selección y la mutación. Obviamente lo que ha pasado es que, el mejor individuo es el mejor que se haya generado al principio, y la población se mantiene estática al no tener forma de alterar ningún individuo.

Efectos de la probabilidad de mutación



Para este caso, hemos elegido una población muy pequeña, de 10 individuos, que evolucionarán durante 1000 generaciones. Además, intentarán optimizar la función de Michalewicz con n=12, la cual es bastante complicada. En ambas utilizamos los mismos parámetros, salvo la probabilidad de mutación.

Para el primer gráfico hemos utilizado una probabilidad de 10%. Esta es demasiado alta, e introduce demasiado ruido a la población, como podemos observar por la media, la cual se mantiene siempre bastante baja. Esto hace que el resultado al que se llegue no sea muy bueno.

En cambio, en el segundo utilizamos una probabilidad de mutación del 2%. Esto hace que haya mucho menos ruido, la media de la población mejore, y se consigan mejores resultados.