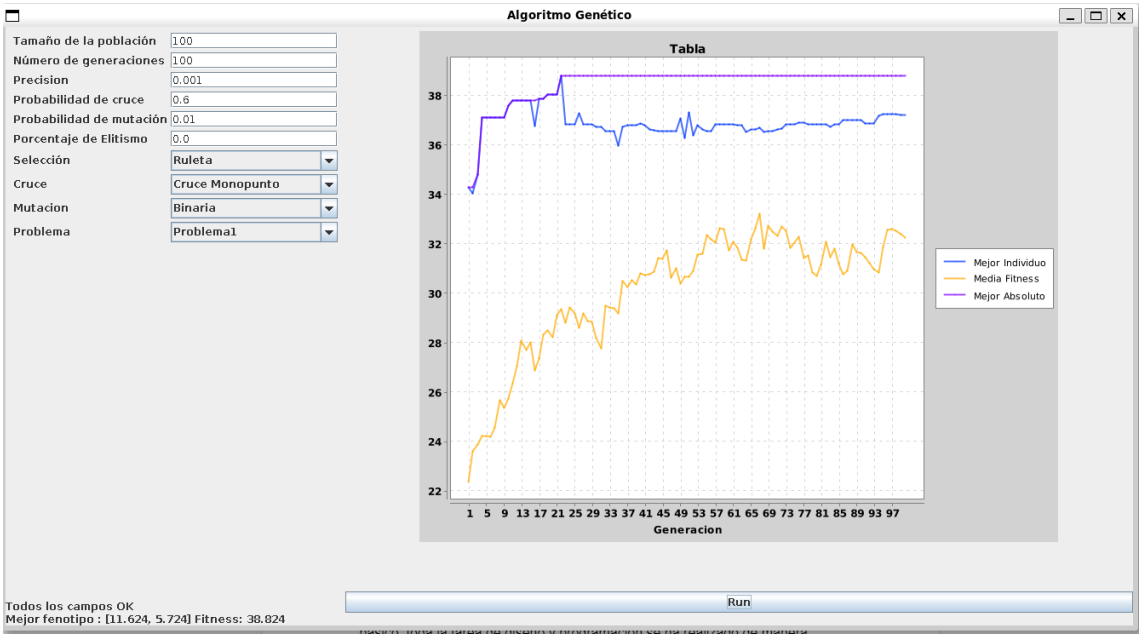
Programación Evolutiva

Practica 1

Ignacio Sánchez Santatecla
David Montoro Couso
Grupo 07

Gráficas:

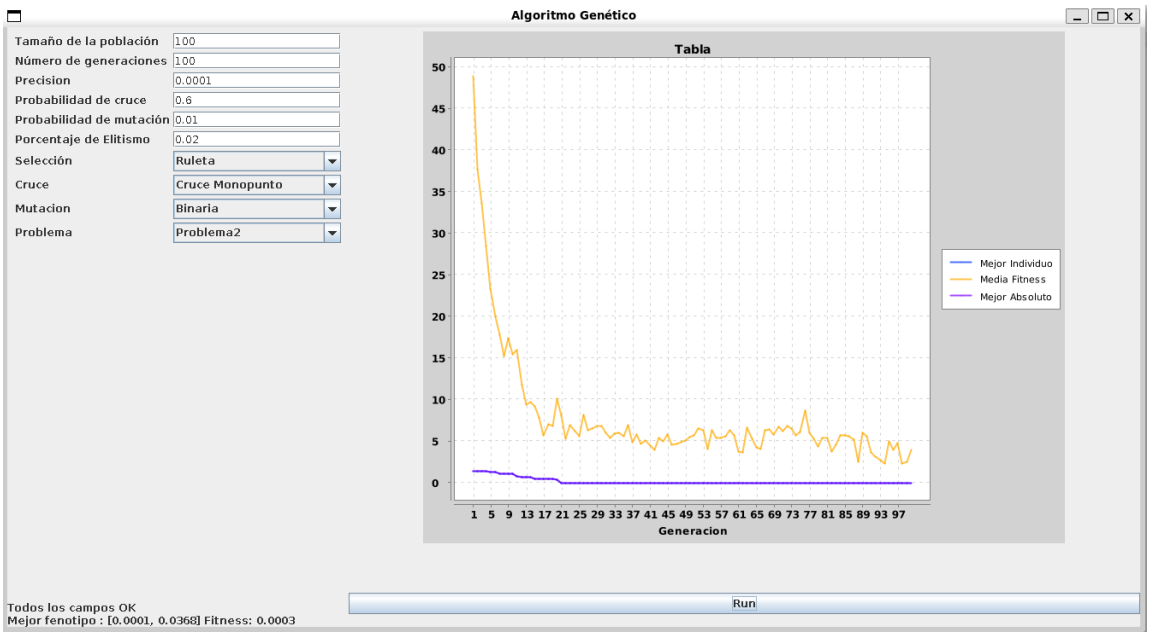
Problema 1:



Conclusión:

En este problema jugar con los parámetros no tiene un efecto muy notable. El elitismo sólo acelera la convergencia, esto mismo ocurre si usamos una selección que le de mucha preferencia a los fitness altos como el truncamiento.

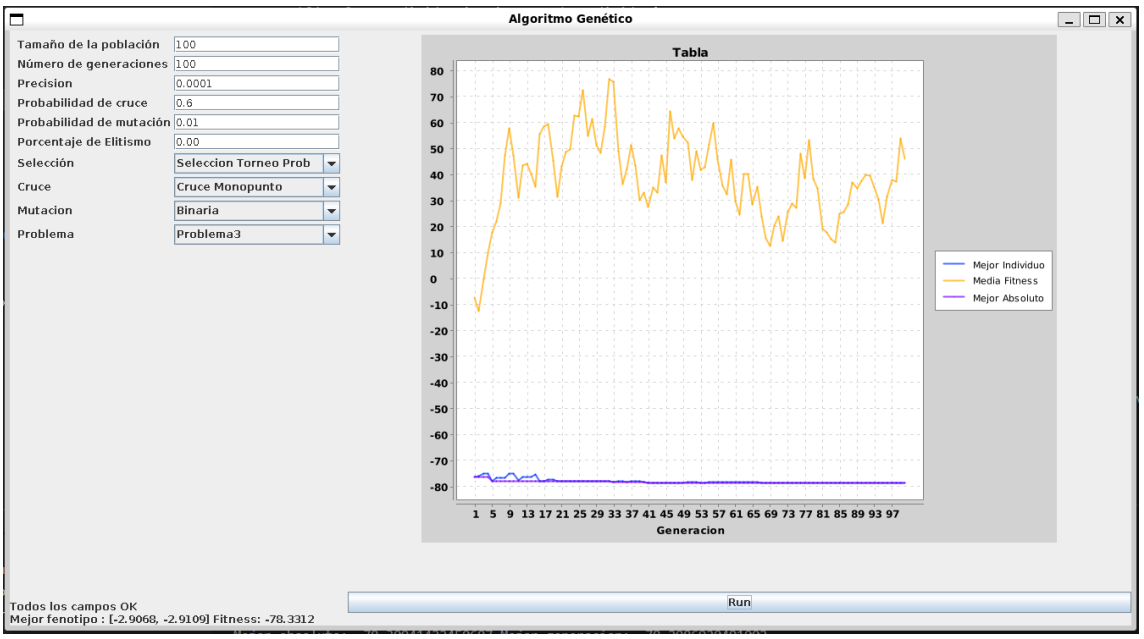
Problema 2:



Conclusión:

En este problema hemos usado un elitismo del 2% para acelerar la convergencia y sobretodo asegurarnos de que si en algún caso el fenotipo de un individuo es cercano a cero en algún parámetro, lo guardemos. Debido a que el rango de números es tan grande y que con valores cercanos a cero el fitness es muy cercano a cero, queremos asegurarnos de que se guarde el mejor individuo. Se puede observar el uso de elitismo en el solapamiento de la gráfica de mejor individuo de cada generación y el mejor individuo total.También hemos aumentado la precisión.

Problema 3:

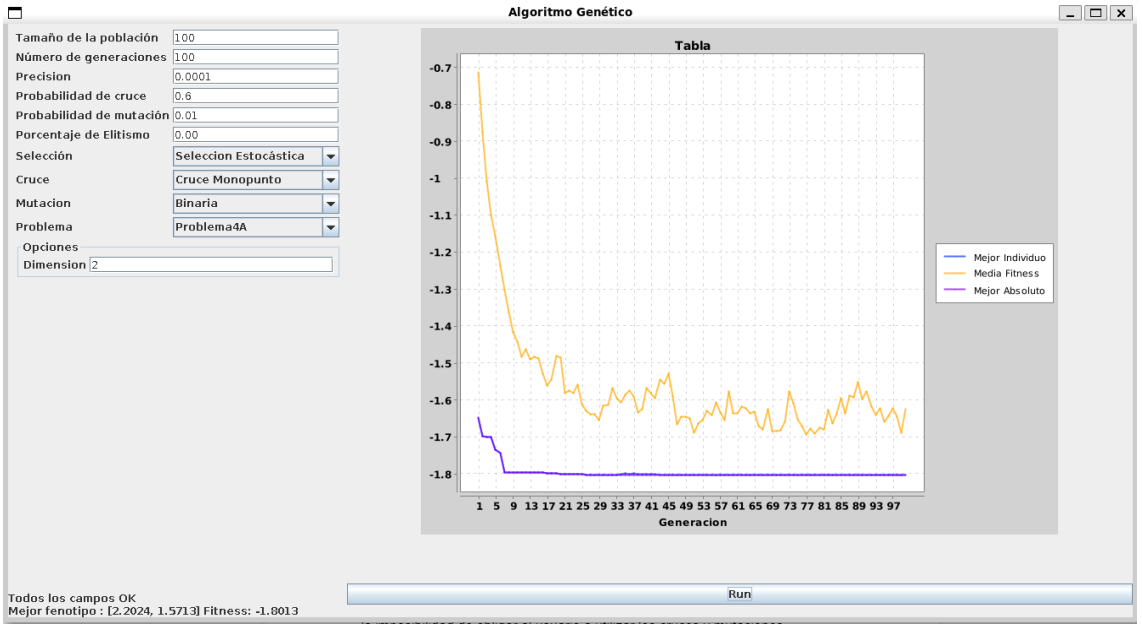


Conclusión:

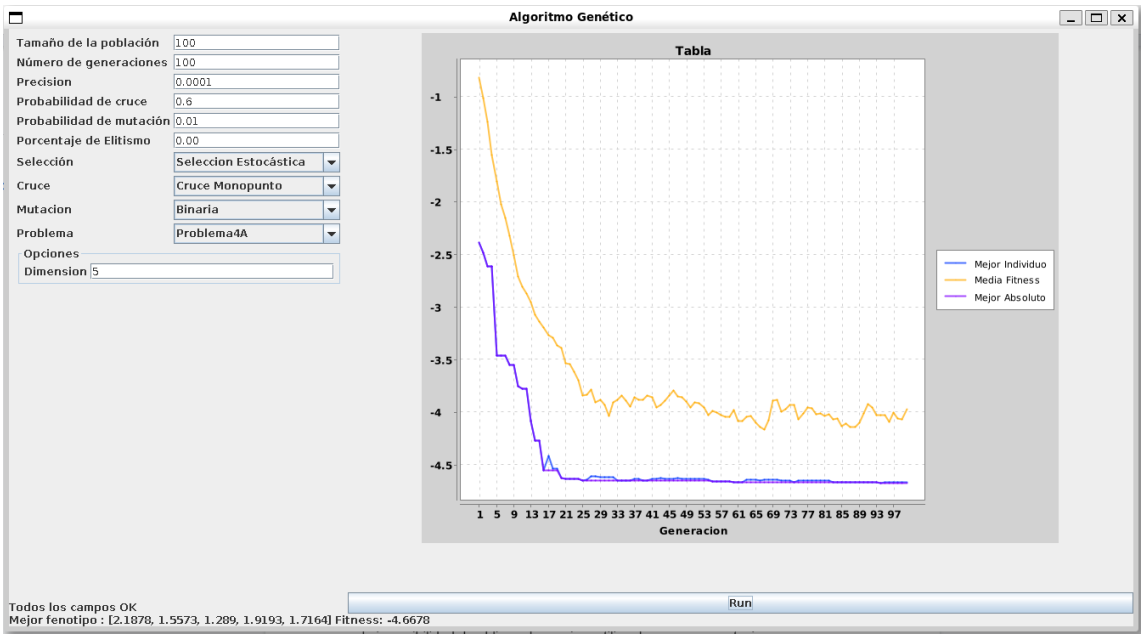
En este problema hemos usado la selección de Torneo Probabilístico, se pueden ver dos efectos claros de esta selección, el primero es la variación que hay en la gráfica de la media dependiendo si ha escogido la mayoría de individuos cercanos a la elite o cercanos a los peores, el segundo es un efecto muy parecido al elitismo debido a que casi con certeza (ya que tiene una probabilidad del 50%) va a escoger al mejor de la generación y este se va a ir propagando.

Problema 4A:

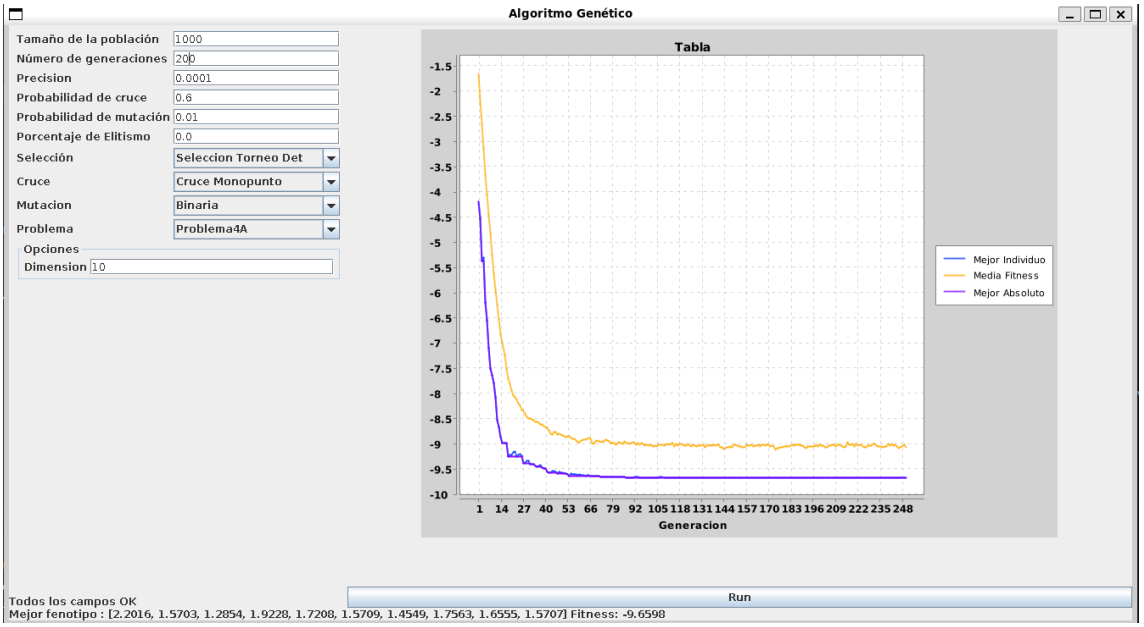
Dimensión 2:



Dimensión 5:



Dimensión 10:



Conclusión 4A:

- Dimensiones 2 y 5:

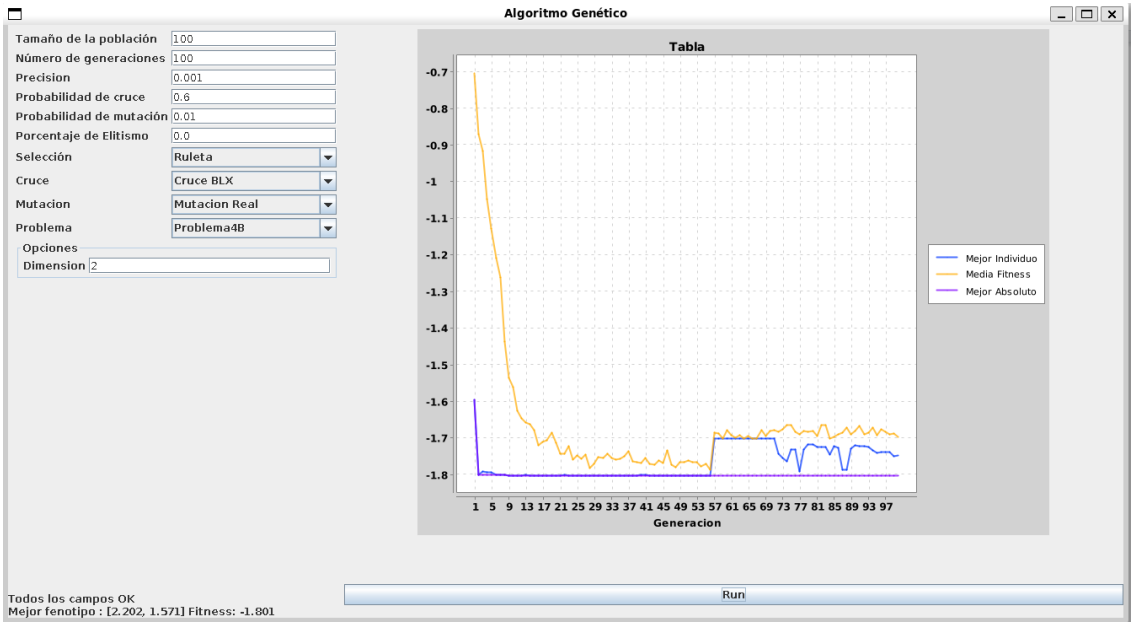
  Hemos usado la selección estocástica, debido a esto se observa un efecto parecido al elitismo (más notable en la dimensión 2) que acelera la convergencia del valor notablemente.

- Dimensión 10:

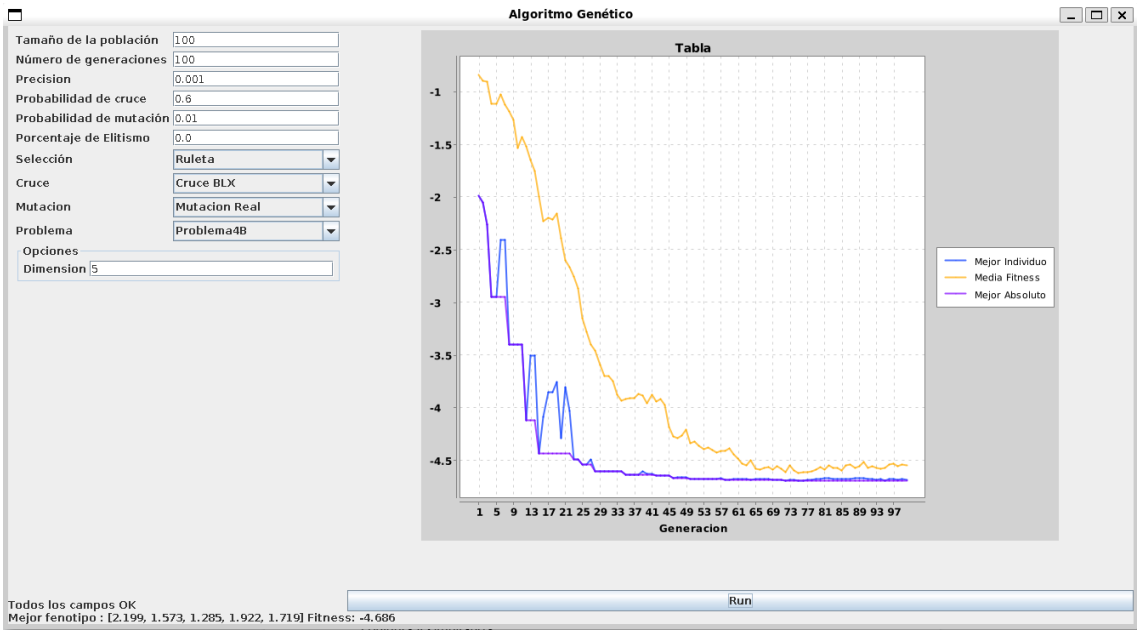
  En este caso como le cuesta mucho más encontrar la solución hemos usado una selección muy agresiva para fomentar que siempre pasen los mejores, el torneo determinista. También hemos aumentado el número de individuos para tener mas posibilidades de exploración y el número de generaciones. Debido a la naturaleza de la selección se consigue el mismo resultado que usando elitismo, no obstante se ve una convergencia más rápida y agresiva ya que el porcentaje de élite que se pasa a la siguiente generación es muy elevado.

Problema 4B:

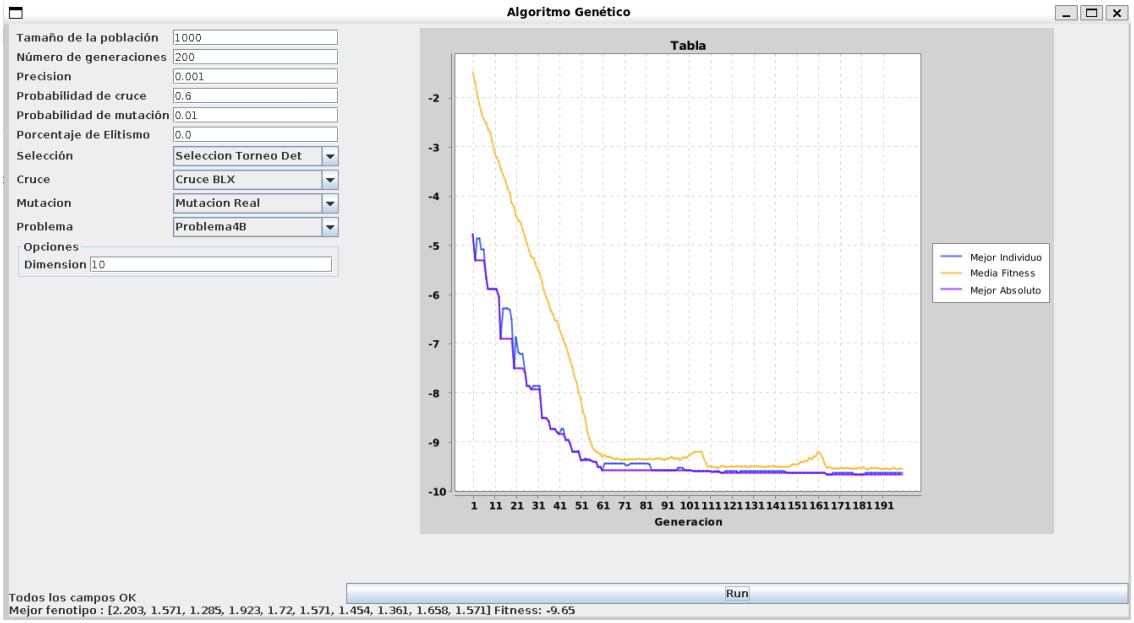
Dimensión 2:



Dimensión 5:



Dimensión 10:



Conclusiones 4B:

- Dimensiones 2 y 5:

  Se puede ver un efecto parecido al problema 4A, siendo la única diferencia que el proceso de selección que hemos escogido es ruleta, lo cual retarda más la convergencia de nuestro algoritmo.

- Dimensión 10:

  Al igual que en el 4A, hemos usado una selección agresiva (torneo determinista) para acelerar la convergencia y en este caso hemos aumentado el número de generaciones a 200 y el número de individuos a 1000 para asegurarnos que siempre da la mejor solución.

Reparto de Tareas

Al tratarse de la primera práctica, donde hemos construido el esqueleto principal, y aprendido los esquemas y el funcionamiento del algoritmo genético básico, toda la tarea de diseño y programación se ha realizado de manera conjunta y simultánea.

Detalles de la implementación

GUI: Hemos decidido utilizar la biblioteca auxiliar para crear formularios que nos ha sido proporcionada en el campus para crea el panel de control del algoritmo. Debido a esto nos hemos topado con algunas limitaciones, como son la imposibilidad de obligar al usuario a utilizar los cruces y mutaciones apropiados para cada tipo de problema o individuo a tratar, o el bloqueo del formulario durante la ejecución para no cambiar parámetros durante la ejecución y alterar el progreso natural del algoritmo, llegando incluso a la detención repentina del algoritmo por las incompatibilidades entre cruces y mutaciones mencionadas previamente.
A cambio, como la vista no se bloquea, podemos ver como la gráfica va evolucionando según va iterando en cada generación.

Individuos y problemas: Hemos decidido generalizar los individuos a dos tipos: binario y real. De esta manera podemos reutilizar esta representación interna que tienen para otros problemas independientemente del nuevo objetivo que tenga. Esto hace que lo que tengamos sea una clase distinta para cada tipo de problema, que se encarga de evaluar el fitness, (a nuestro parecer es lo más óptimo ya que es el problema el que determina como de bueno es un individuo según su fenotipo), almacenar la dimensión, los intervalos mínimos y máximos y de determinar si se trata de un problema de maximización o minimización

Anotaciones de uso: El panel de control está al lado izquierdo donde se seleccionan los parámetros del algoritmo genético. En la parte inferior está el botón de inicio. Si se quiere hacer zoom solo hay que pinchar en el inicio del intervalo a ampliar, arrastrar el ratón y soltar al final del intervalo.

Hemos trabajado con Java 17, cualquier versión anterior puede dar problemas.

Puede ser que sea necesario incluir la librería que genera la gráficas, para ello, usando Eclipse lo que tienes que hacer es incluir la librería en el Build Path, la librería está alocada en la carpeta libs, el proceso es el siguiente:

Click derecho proyecto -> Properties -> Java Build Path -> ClassPath -> Add Jars -> seleccionas la librería desde la carpeta -> apply and close