**Memoria programación evolutiva.**

**Grupo 3**

Carlos Lozano Casado.

Erick Eresi.

**Índice**

-Introducción

-Código

-Técnicas

-Graficas y análisis

**-Código:**

Nuestro código está centrado en tres grandes clases:

1º- AlgoritomoGenetico: esta es la clase encargada de inicializar la población e ir ejecutando paso a paso el algoritmo, además de guardar ciertos datos interesantes para el algoritmo. Desde un cierto punto de vista actúa como el controlador en un modelo-vista-controlador, ya que es el que obtiene los datos para actualizar la vista, y llama al modelo para ejecutar acciones.

2º-Poblacion: esta clase abstracta guarda todos los cromosomas y los selecciona para que se crucen y elegir la nueva población. De esta clase heredan varias clases de población que implementan las distintas selecciones de la nueva generación. Esta clase es la que va llamando al cruce y la mutación de los cromosomas. Hay que comentar que la selección por truncamiento, pide un parámetro adicional que es el porcentaje de población distinta que se queda el algoritmo.

3º-Cromosoma: esta clase abstracta en la que almacena la información relevante del cromosoma: su codificación, aptitud, fenotipo, etc. Tiene un nivel de abstracción para el tipo de codificación del cromosoma, si es binario, real, etc.; y otro nivel de abstracción que es la implementación de un problema concreto, de modo que en esta práctica la función 5 tiene dos clases de cromosoma, una para una implementación con booleanos, binario puro, y otra con números reales. La idea de estos niveles de abstracción es que cada uno implemente el cruce y la mutación de forma personalizada para una codificación y problema.

Hay otras clases pero no son tan relevantes para el algoritmo, esta son las factorías, las vistas, y otra clases auxiliares.

Hemos procurado pensar a futuro y tener todo para que sea lo más readaptadle posible, de modo que para nuevos problemas solo hay que crear un nuevo cromosoma, y adaptar las factorías, en el peor de los casos modificar el cromosoma.

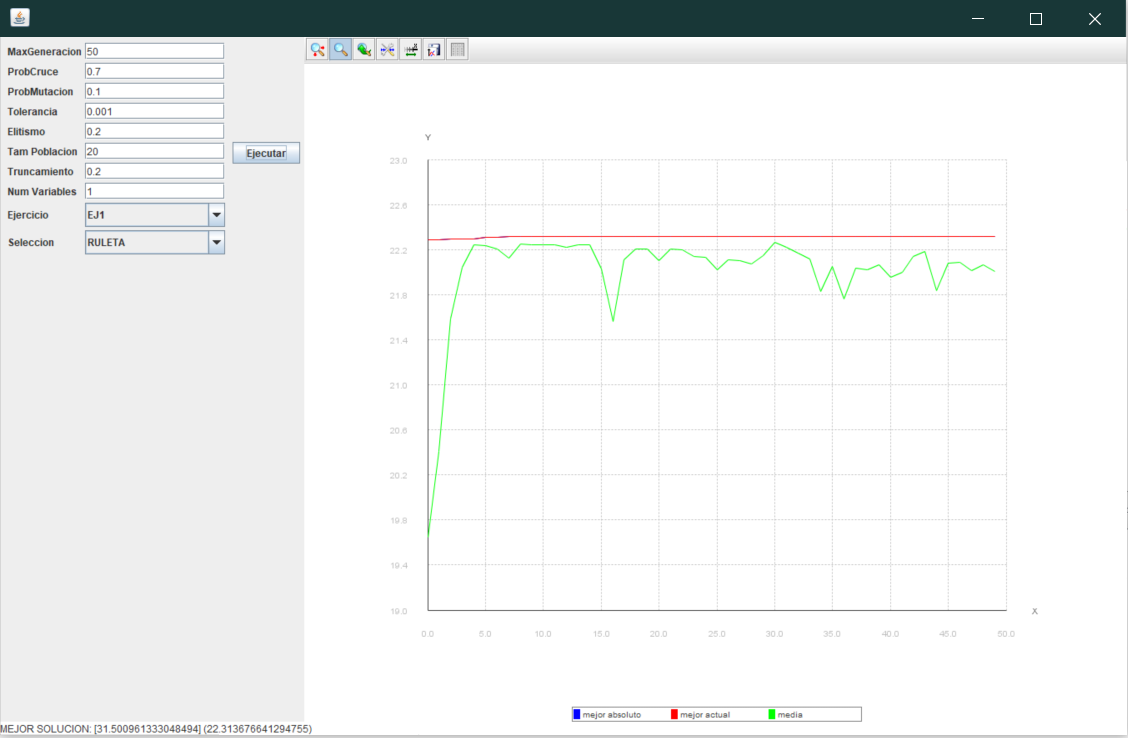
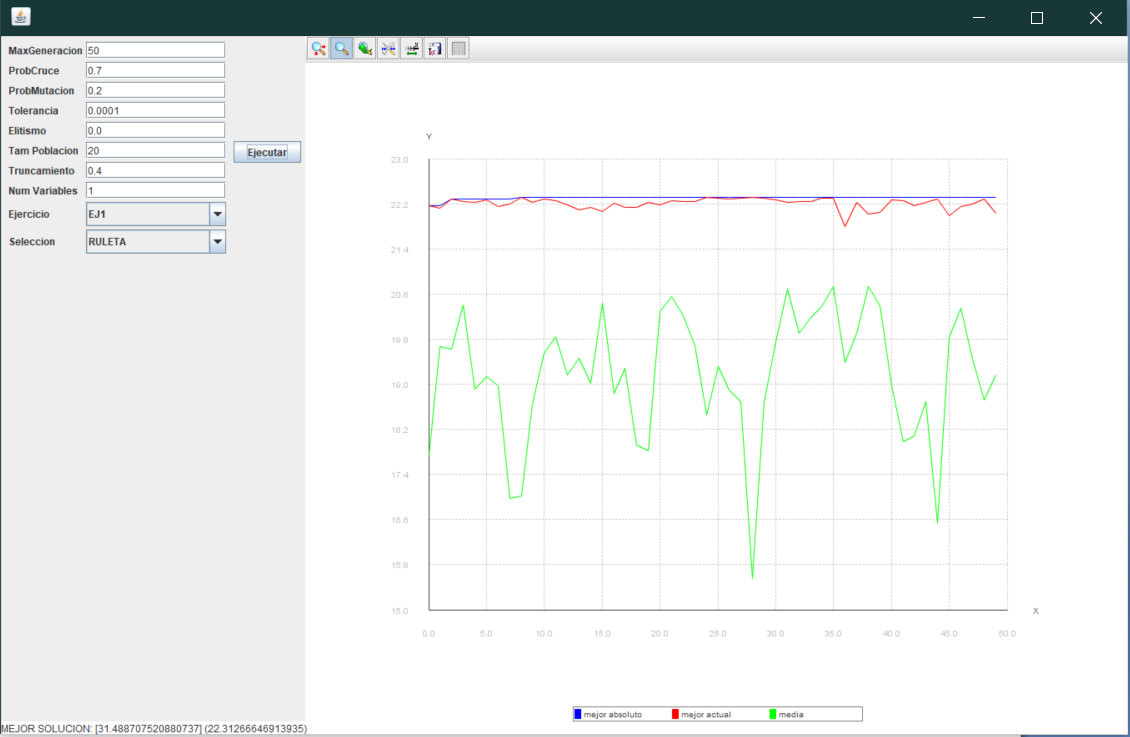
**-Técnicas:**

Hemos aplicado las técnicas vistas en clase, tal vez remarcable el uso de elitismo, que hemos implementado manteniendo un array de los mejores cromosomas y los volvemos añadir a la población después de la mutación, y actualizamos esta array. De este modo no perdemos las mejores soluciones que encontramos, cosa que nos pasaba si no estaba implementada.

**-Graficas y análisis:**

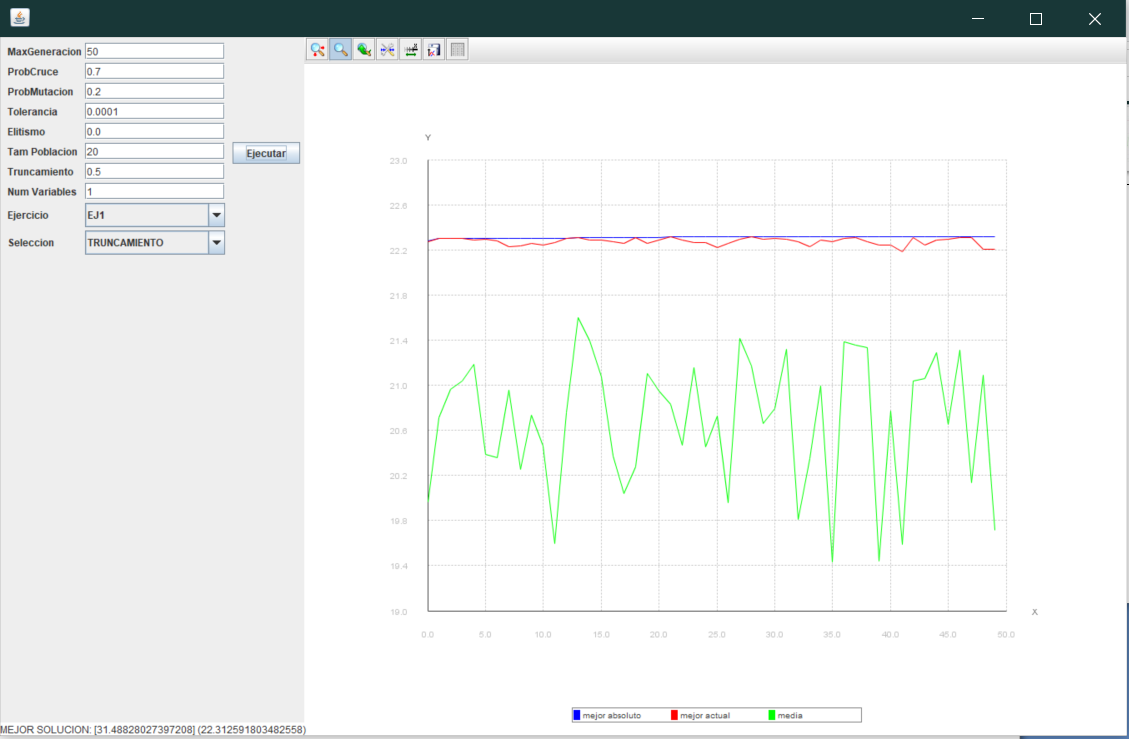
Empiezo con una análisis mas especifico de cada problema, y termino con uno mas general.

-P1: En este primer problema se puede ver bien el efecto de tener elitismo, en la primera imagen, el mejor histórico y el actual del algoritmo se solapan, mientras que sin elitismo, aunque se mantiene por lo general cerca del mejor, llega a variar bastante, teniendo en cuenta la precisión dada (0.0001).

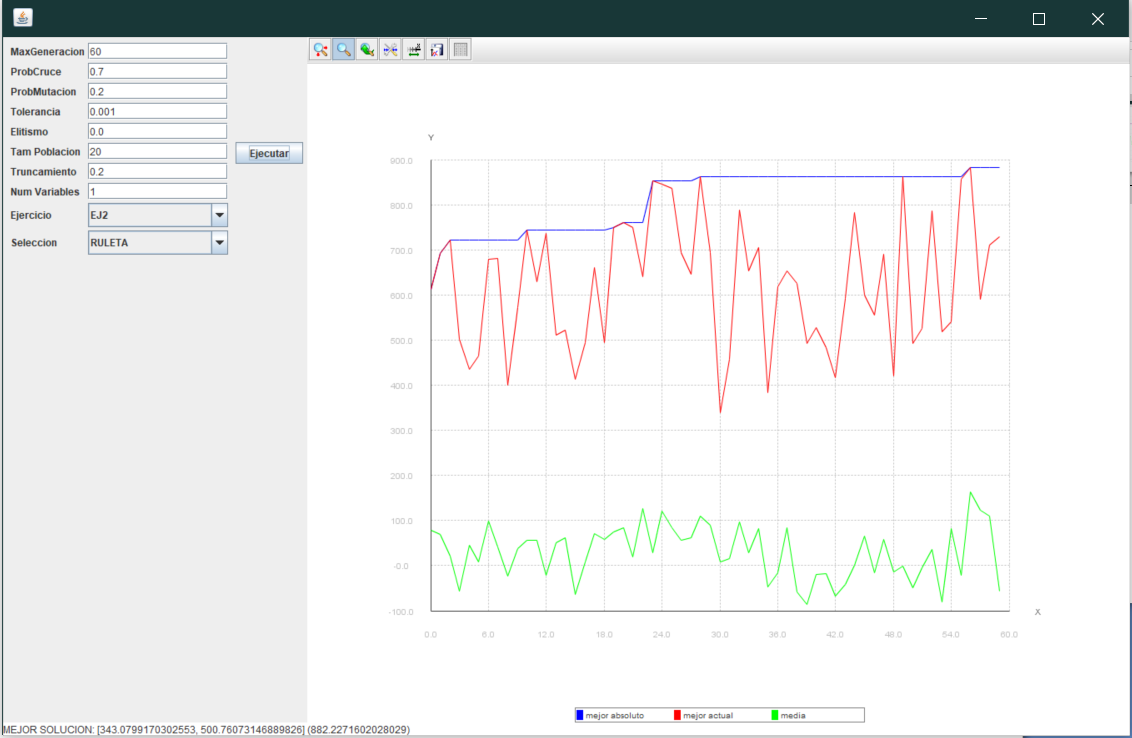


Con elitismo Sin elitismo

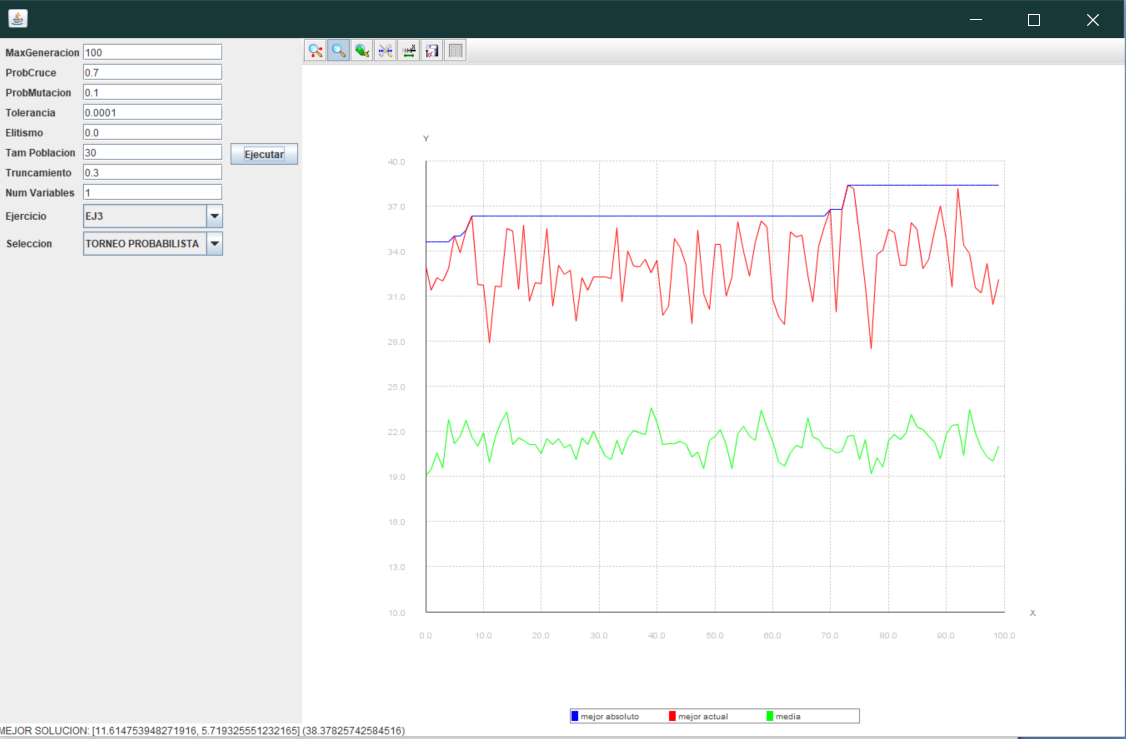
De este ejercicio como mucho comentar, que quitando el elitismo, el que mantiene un mejor mas cercano al histórico es por selección por truncamiento.



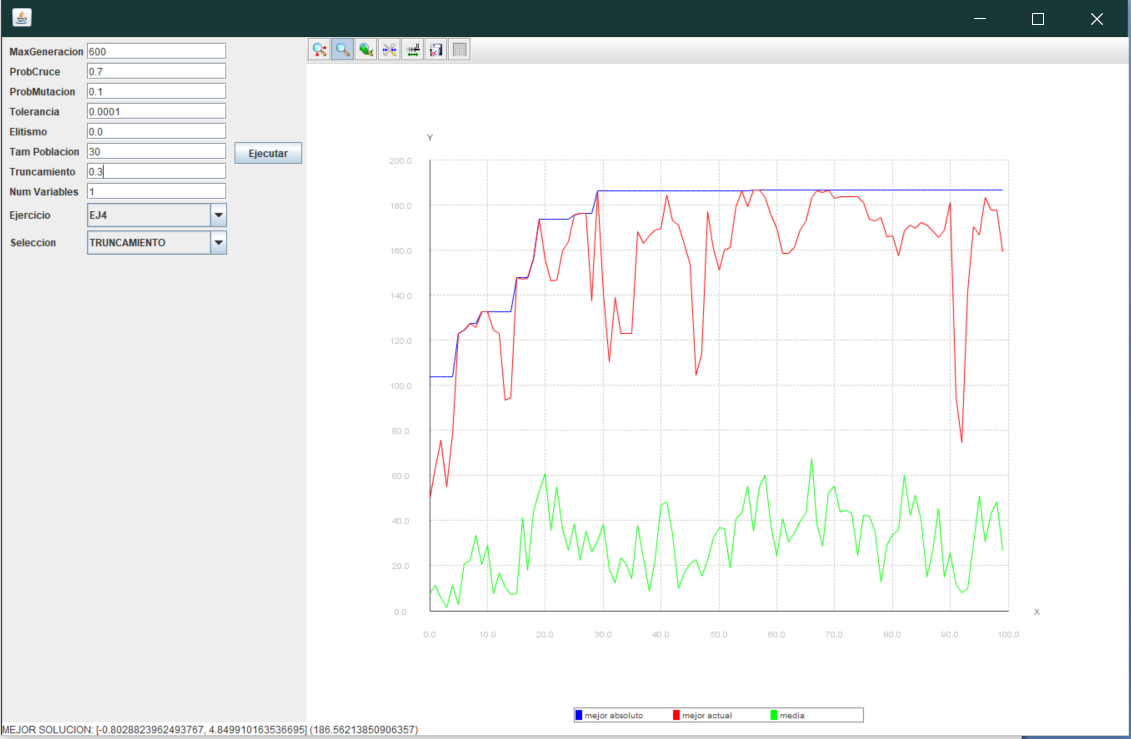
-P2: De este ejercicio lo que más nos llama la atención es lo que tarda en encontrar el mejor llegando a cambiar este varias veces durante el trascurso del algoritmo, comparado con el anterior que lo encontraba el mejor o uno muy cercano en muy pocas generaciones. Y que él no elitismo en este problema es incluso más exacerbado que en el anterior. También comentar la gran diferencia entre la media y el mejor.



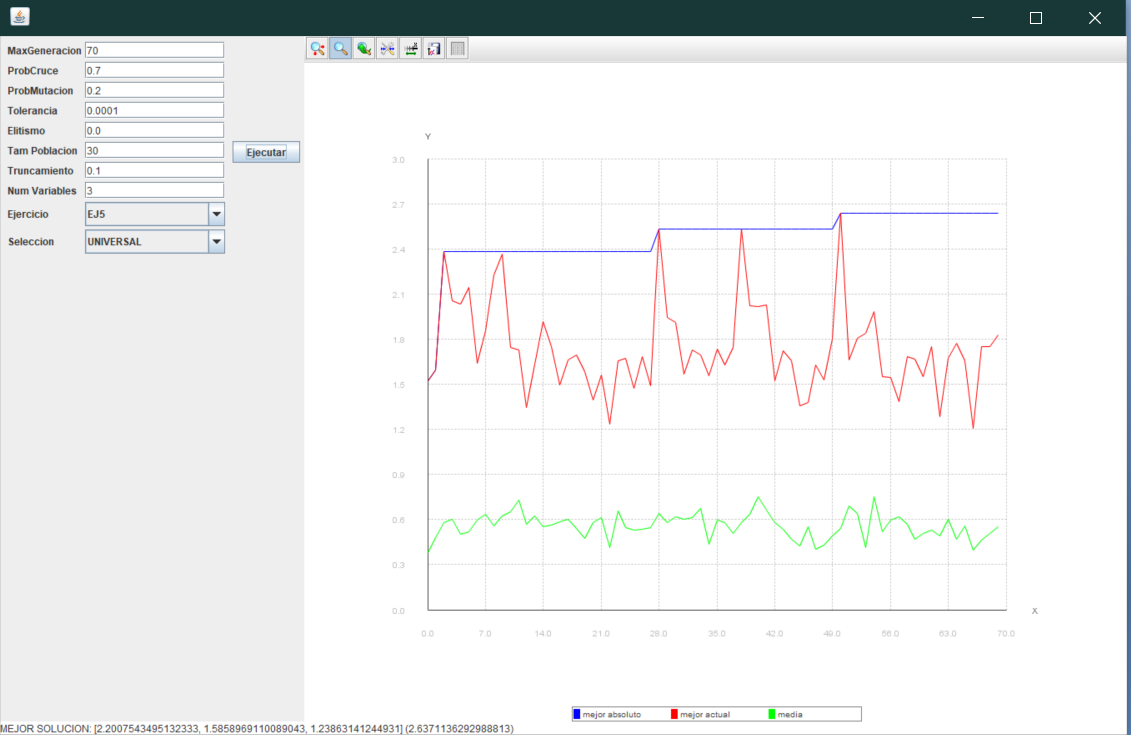
-P3: De este hay poco que comentar ya que las graficas salen muy parecidas al del anterior, como mucho que la media es más cercana a los valores mejores.



-P4: Este también es muy variable, incluso con truncamiento, con un valor un poco mayor del 25%, varía mucho, para valores menores de truncamiento, no le afecta tanto, y aun así comparado con otros problemas varía mucho con esos valores.



-P5: con una variable se comporta como el ejercicio 1, encuentra rápidamente el valor óptimo, y luego no varía mucho el histórico del actual. De 2 a 3 variables da resultados más parecidos al resto de graficas, a veces le cuesta encontrar el mejor, pero lo consigue. Con 4 o más, no suele encontrar el mejor.



En general cambiar los valores del algoritmo (ProbCruce, de mutacion, tolerancia, ect.) no produce cambios a menos que sean muy extremos, lo mismo ocurre con las selecciones, salvo truncamiento, y este con valores altos (mayores o iguales al 0,3), no producen por lo general cambios apreciables en el algoritmo; tal vez cambie como varié el mejor actual o la media, pero por lo general producen resultado muy parecidos. La excepción como ya he dicho es truncamiento, con valores alrededor de 0,2 (20%), reduce muchísimo la varianza en el mejor actual en casi todos los casos.

Muchas generaciones o mucha población no producen grandes cambios, pero si obviamente poner muy pocas. El elitismo con valores mayores a 0 se activa, dando lugar a, como ya se ha mencionado antes, que el mejor histórico y actual se solapen, también eleva la media, y me parece que acelera el proceso de encontrar un optimo, pero no estoy del todo seguro, es muy sutil.