Programación Evolutiva
Práctica 3

Roberto Plaza Hermira

Gorka Silva Ramón

Detalles de implementación

1. Gramática Evolutiva

Adaptación del estilo de las gramáticas. Hemos adaptado las gramáticas de ejemplo para poder reconocerlas fácilmente haciendo cambios como:

- En el ejemplo era: (<expr> AND <expr>). Y ahora es: AND (<expr>) (<expr>)
- Ajustes en la disposición de los paréntesis para que todas las gramáticas tengan el mismo formato.

Cruces y mutaciones implementadas.

Cruce OX	Mutación por inserción
Cruce OX-PP	Mutación por intercambio
Cruce CO	Mutación por inversión
	Mutación heurística

Guía de uso.
Para el multiplexor 4 a 1 se pueden utilizar las gramáticas: gramatica1.txt, gramatica3.txt.
Para el multiplexor 8 a 1 es necesario utilizar la gramática: gramática2.txt.

2. Programación Genetica

Implementación del árbol:

El árbol consiste de una clase abstracta “Arbol” completada por herencia con las clases “Nodo” y “Hoja”. La clase “Nodo” agrupa a las expresiones funcionales mientras que la clase “Hoja” agrupa a las expresiones terminales.

Las clases que heredan de “Hoja” son: “HojaA0”, “HojaA1”, “HojaA2”, “HojaD0”, “HojaD1”, “HojaD2”, “HojaD3”, “HojaD4”, “HojaD5”, “HojaD6” y “HojaD7”.

Las clases que heredan de “Nodo” son: “NodoAnd”, “NodoNot”, “NodoIf” y “NodoOr”.

Cada “Arbol” contiene los diferentes datos:

- Lista de hijos
- Padre
- Profundidad: esta variable indica la profundidad máxima de los hijos. Ejemplo: si la profundidad indicada en la interfaz es de 5, la raíz del árbol tendrá profundidad 4, mientras que las hojas tendrán como mínimo 0.
- Niveles hijos: esta variable indica la mayor altura de entre sus hijos. El propio nodo no está incluido. Tiene un valor máximo igual a “Profundidad”.
- Tamaño subárbol: esta variable indica el tamaño del subárbol del que se es la raíz.
- m6: es un boolean que indica si se tiene que ejecutar para un multiplexor de 6 entradas (true) o para uno de 11 entradas (false).

Cruces y mutaciones implementadas.

Cruce por intercambio de subárboles

Mutación por Expansión
Mutación por Contracción
Mutación Funcional
Mutación Hoist
Mutación Terminal
Mutación por Árbol
Mutación por Permutación

Gramática Evolutiva

1. Multiplexor 4 a 1. Fitness máximo: 64

Este problema no encuentra resultados significativos con tamaño de población = 100 y número de generaciones = 100, independientemente de la longitud del cromosoma y del número de wraps.

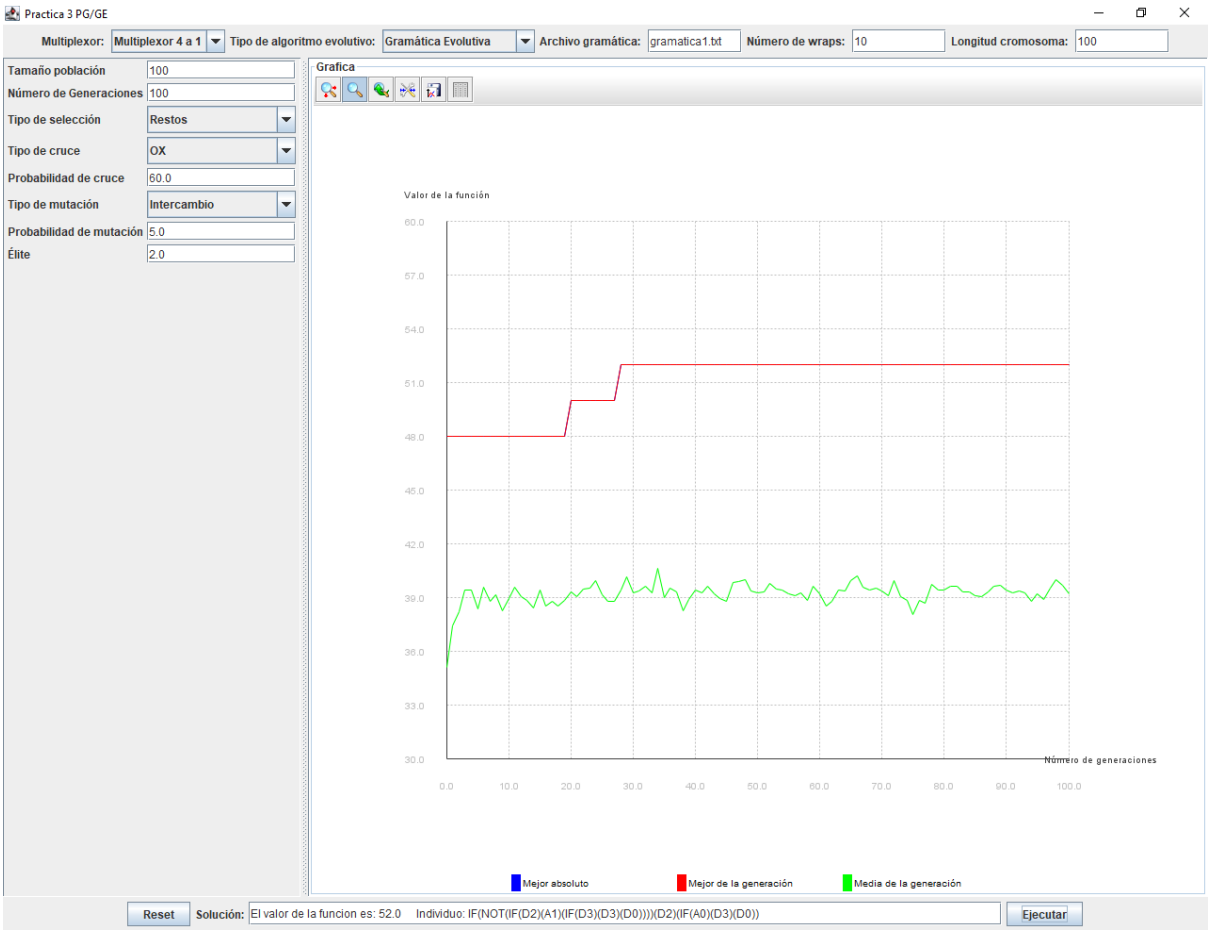

Imagen1:
Número de wraps = 10.
Longitud del cromosoma = 100.
Tamaño de población = 100.
Número de generaciones = 100.

Cuando utilizamos tamaño de población = 1000 y número de generaciones = 300 empezamos a encontrar resultados mejores, pero aun así no llega a la solución del problema. El máximo alcanzado en varias ejecuciones ha sido 56.

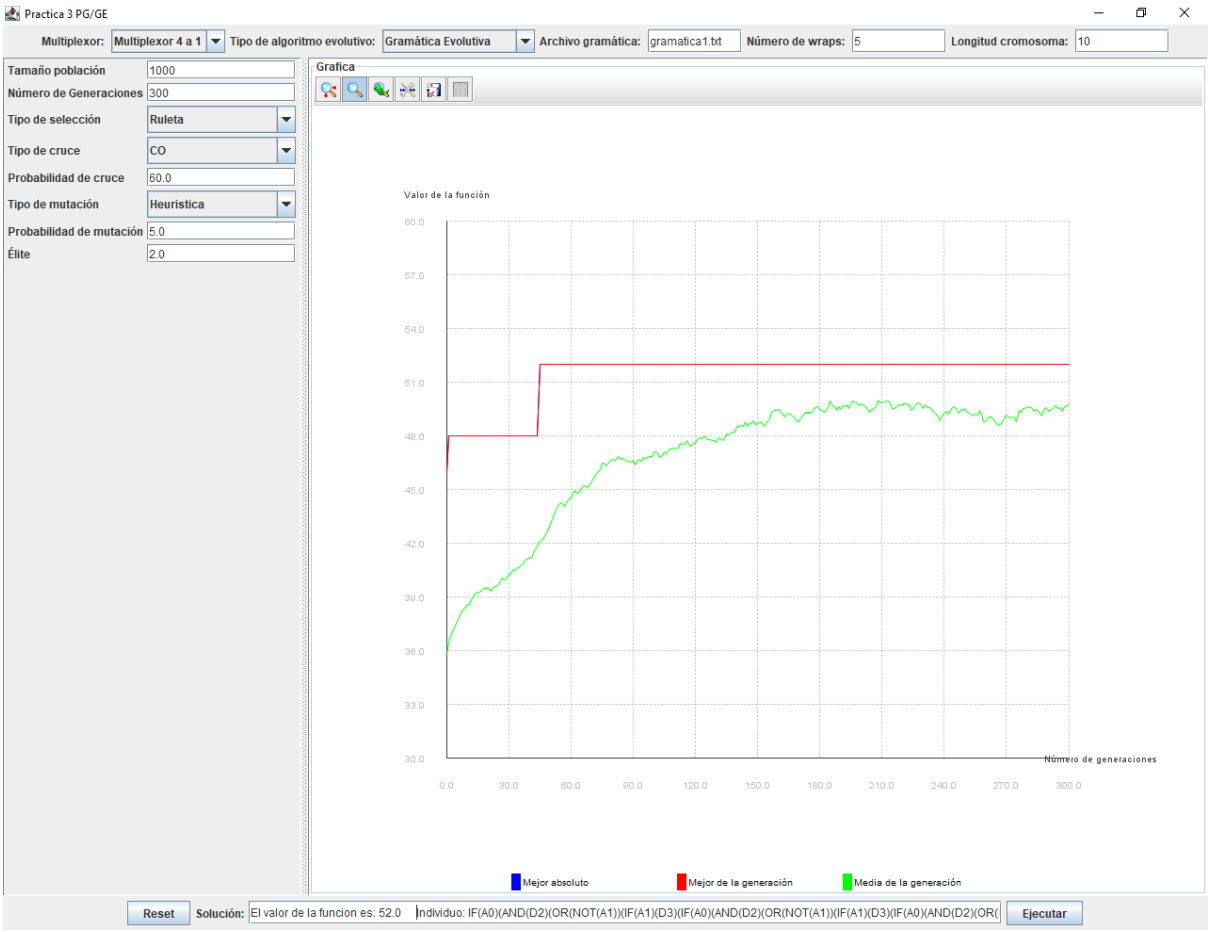

Imagen2:
Número de wraps = 5.
Longitud del cromosoma = 10.
Tamaño de población = 1000.
Número de generaciones = 300.

En general todas las combinaciones de cruces, mutaciones y selecciones dan resultados parecidos. Aunque el cruce CO pueden conseguir peores resultados cuando el primer mejor global es un individuo cuya expresión es bastante corta. Esto hace que el resto de individuos sean parecidos a este, y por lo tanto, que el algoritmo converja muy rápido en una solución con poco fitness.

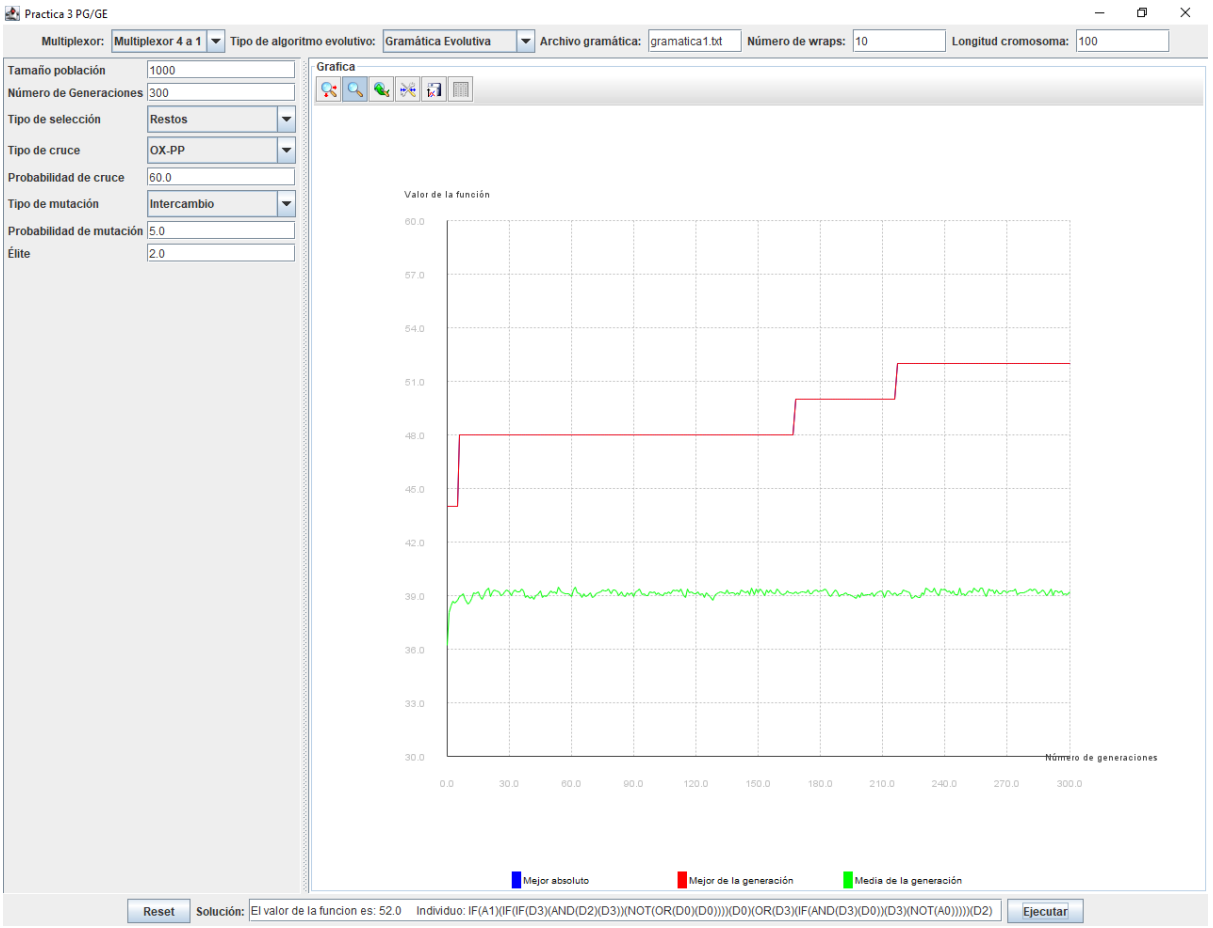

Imagen3:
Número de wraps = 10.
Longitud del cromosoma = 100.
Tamaño de población = 1000.
Número de generaciones = 300.

Con un número mayor de wraps y de longitud del cromosoma se pueden obtener una ligera mejora en los resultados en general, pero el máximo sigue siendo 56.

2. Multiplexor 8 a 1. Fitness máximo: 2048

Imagen1:
Número de wraps = 5.
Longitud del cromosoma = 15.
Tamaño de población = 1000.
Número de generaciones = 300.

Imagen2:
Número de wraps = 8.
Longitud del cromosoma = 25.
Tamaño de población = 500.
Número de generaciones = 150.

Imagen3:
Número de wraps = 15.
Longitud del cromosoma = 75.
Tamaño de población = 100.
Número de generaciones = 100.

Reparto de tareas

Roberto se ha encargado principalmente de:

- Programación Genética

Gorka se ha encargado principalmente de:

- Gramática Evolutiva
- Adaptación Interfaz GUI
- Debug

La memoria se ha hecho en cooperación.