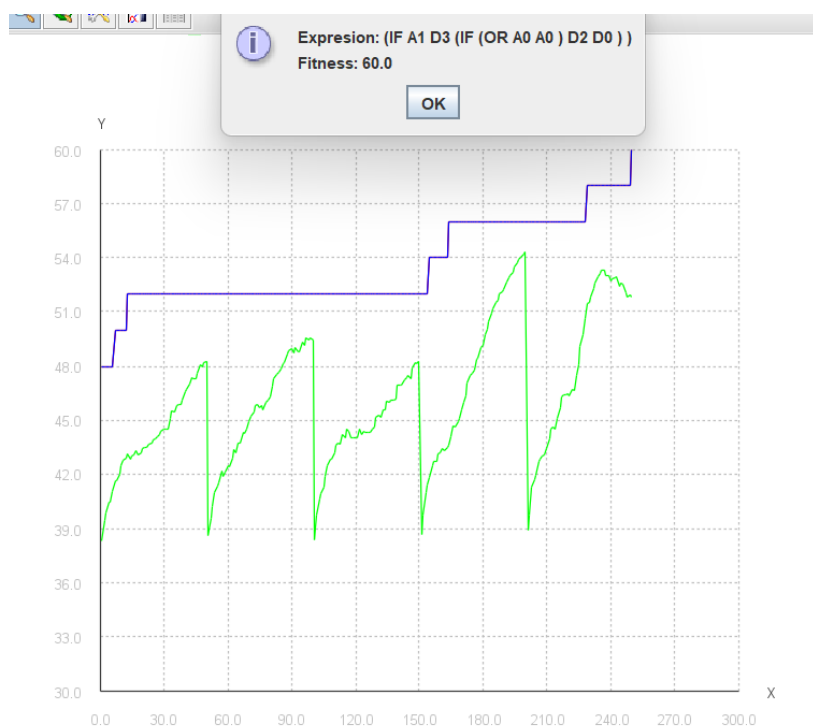


Práctica 3

Gramática evolutiva para generación de un multiplexor.



Elena Novillo Luceño

Lucía Heredero Illán

Grupo 11.

Resultados de ejecuciones:

Multiplexor de 2 a 4: el multiplexor en cuestión cuenta con 2 entradas (A0 y A1) y 4 salidas (D0, D1, D2 y D3). La mejor aptitud encontrada con la gramática elegida ha sido 60.0 (muy cercano a 64, el máximo posible). Si nos fijamos en la gráfica (*gráfica 1.1*) en la que obtenemos ese resultado, podemos observar que hemos aplicado selección por ruleta, cruce monopunto, o wraps, elitismo en un 2% y porcentajes de cruce y mutación del 80% y del 90% respectivamente. Creemos que este resultado (pese a no ser muy común) se encuentra debido al elitismo. Si ejecutamos con unos valores parecidos, aunque sin elitismo, observamos (*gráfica 1.2*) que no obtenemos el fitness 60, sino tan solo 56. Pese a que en gramáticas evolutivas, la mutación y la aleatoriedad afectan positivamente, para poder obtener el mejor resultado (para nuestra gramática) es necesario introducir elitismo y para aumentar la variabilidad utilizar un método de regeneración de la población (explicado más adelante, en el apartado de “Implementación”). Por otro lado, en este mismo caso podemos observar que, al utilizar más de o wraps obtenemos expresiones complejas en exceso (*gráfica 1.3*) que, aunque encuentran un resultado bueno, obtienen muchas veces expresiones redundantes. Habitualmente no se llega al valor 56, si no que queda en torno al 52 y 54 (si no aplicamos muchas generaciones) como podemos observar en la última gráfica (*gráfica 1.4*)

Selección de parámetros

Tamaño de la población:

500

Números de generaciones:

250

Porcentaje de cruce (sobre 100):

90

Porcentaje de mutación (sobre 100):

80

Porcentaje de elitismo (sobre 100):

2

Método de selección:

Ruleta

Método de cruce:

Monopunto

Multiplexor:

Multiplexor2a4

Wraps:

0

Método de mutación: básica

Run

Graficas

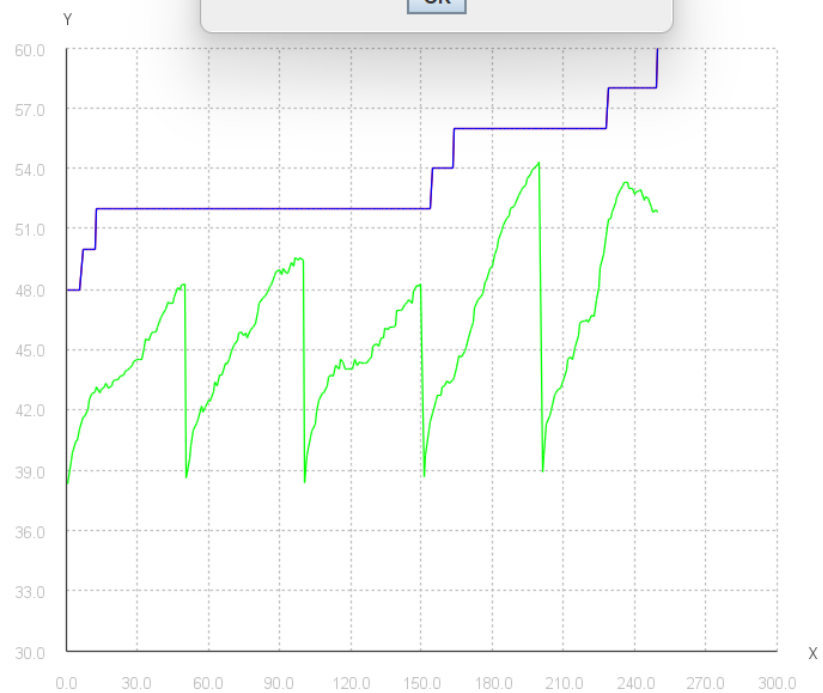


Message

Expresión: (IF A1 D3 (IF (OR A0 A0) D2 D0))

Fitness: 60.0

OK



media mejor mejor absoluto

gráfica 1.1

Selección de parámetros

Tamaño de la población:
500

Números de generaciones:
250

Porcentaje de cruce (sobre 100):
90

Porcentaje de mutación (sobre 100):
80

Porcentaje de elitismo (sobre 100):
0

Método de selección:
Truncamiento

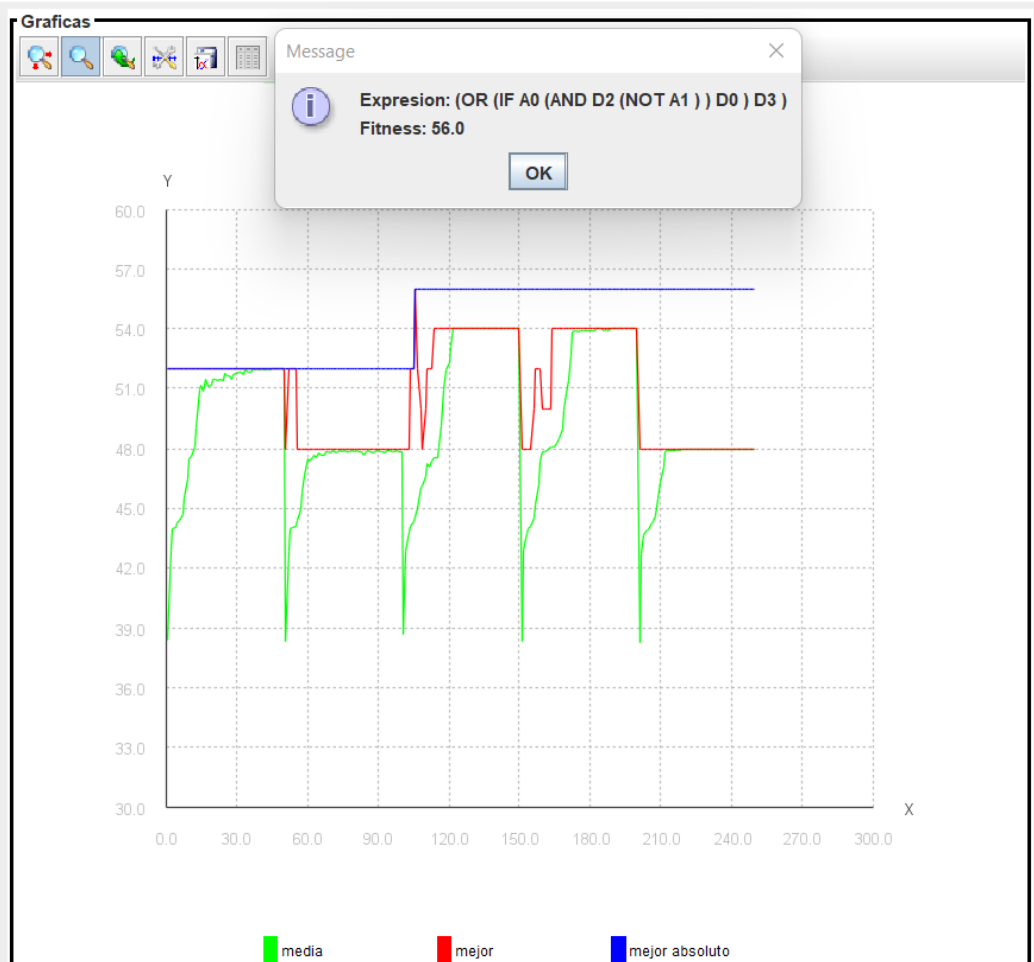
Método de cruce:
Monopunto

Multiplexor:
Multiplexor2a4

Wraps:
0

Método de mutación: básica

Run



gráfica 1.2

Selección de parámetros

Tamaño de la población:
500

Números de generaciones:
100

Porcentaje de cruce (sobre 100):
90

Porcentaje de mutación (sobre 100):
80

Porcentaje de elitismo (sobre 100):
0

Método de selección:
Ruleta

Método de cruce:
Discreto uniforme

Multiplexor:
Multiplexor2a4

Wraps:
0

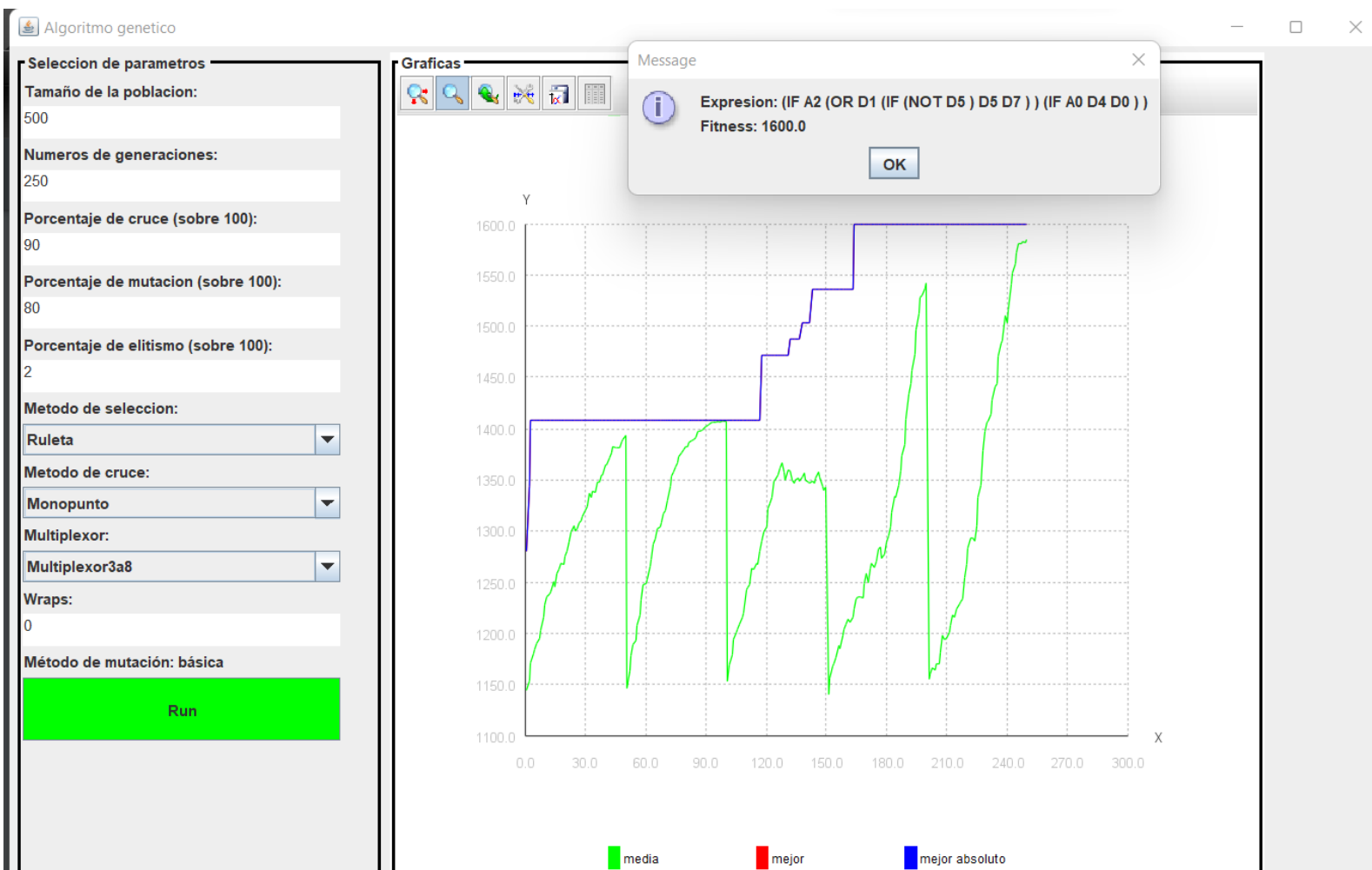
Método de mutación: básica

Run



gráfica 1.4

Multiplexor de 3 a 8: Este multiplexor, al contar con más entradas y salidas, es más complejo de expresar con una gramática y, por tanto, es más complicado obtener el mejor resultado posible (2048.0). Como las gramáticas consisten en llamadas recursivas aumenta mucho el coste en tiempo del algoritmo y si utilizamos muchos individuos o muchas generaciones el programa puede tardar unos minutos en ejecutarse completamente. El caso mejor encontrado al realizar pruebas ha obtenido un fitness de 1600.0 (*gráfica 2.1*) que, aunque se encuentra muy alejado del ideal, es considerablemente mejor que el obtenido en una ejecución habitual, que consigue tan solo 1408.0 (*gráfica 2.2*). Para obtener la mejor ejecución hemos utilizado los mismos parámetros que en la gráfica mejor del apartado anterior. Los resultados en la media podemos conseguirlos aplicando selección por restos, cruce discreto uniforme, un 80% y 90% de porcentaje de mutación y cruce respectivamente, sin elitismo y con 0 wraps (para evitar expresiones demasiado complejas). También encontramos ejecuciones peores, de 1344.0 si aplicamos un menor porcentaje de mutaciones (20%), truncamiento y discreto uniforme como métodos de selección y de cruce, respectivamente (*gráfica 2.3*).



gráfica 2.1

Selección de parámetros

Tamaño de la población:

100

Números de generaciones:

200

Porcentaje de cruce (sobre 100):

90

Porcentaje de mutación (sobre 100):

80

Porcentaje de elitismo (sobre 100):

0

Método de selección:

Restos

Método de cruce:

Discreto uniforme

Multiplexor:

Multiplexor3a8

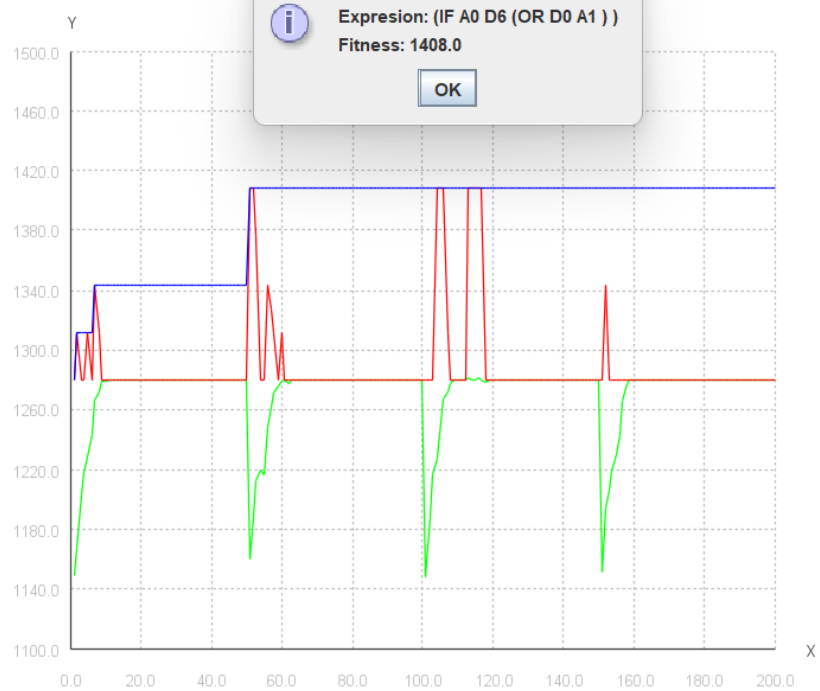
Wraps:

0

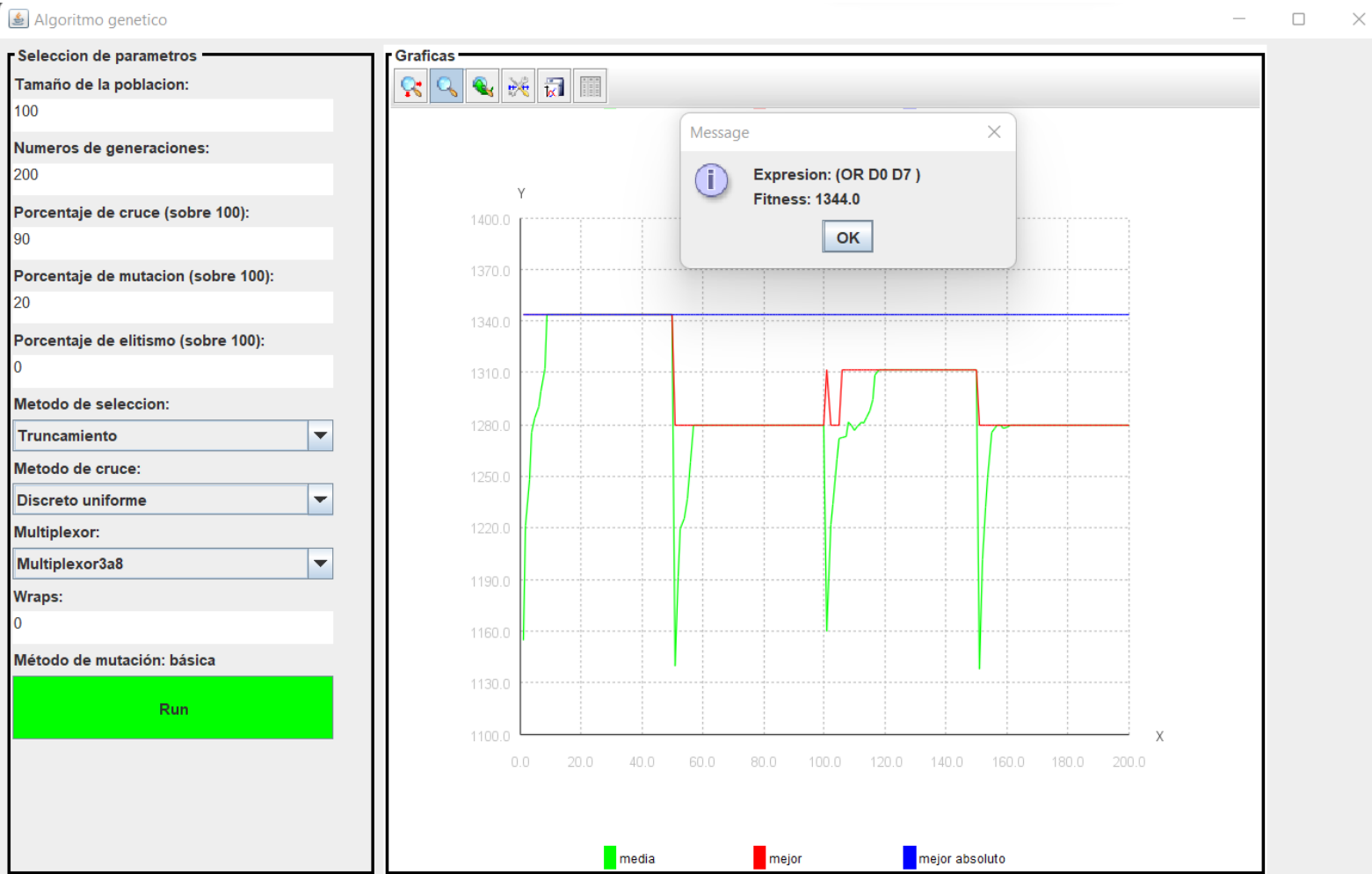
Método de mutación: básica

Run

Graficas



gráfica 2.2



gráfica 2.3

Implementación:

Los métodos de selección, mutación y cruce son los mismos que los vistos anteriormente aunque adaptado para esta representación del cromosoma entero. En concreto son:

- Selección:
 - Ruleta
 - Torneo probabilístico y determinístico
 - Estocástico universal
 - Truncamiento
 - Restos
 - Ranking
- Mutación: mutación básica adaptada para que se sustituya un gen entero por un número aleatorio.
- Cruce:
 - Monopunto
 - Discreto uniforme

Por otro lado, la representación del individuo está en función del tipo del multiplexor. Para el multiplexor de 2 a 4 utilizamos un cromosoma de 10 codones mientras que para el multiplexor de 3 a 8 utilizamos 15. También se tiene que el número que puede tomar cada codón es un número en el intervalo [0-255].

La gramática propuesta se corresponde con la primera que aparece en el anexo de la práctica. Para garantizar la terminación de una expresión, si nos quedamos sin wraps y todavía no hemos acabado, generamos terminales aleatorios hasta que finalice.

Para aumentar la variabilidad hemos utilizado regeneración de la población, es decir, cada cierto tiempo (50 generaciones), se regeneran todos los individuos de la población, teniendo en cuenta que si estábamos aplicando elitismo, la élite de las antiguas generaciones se reinserta en la nueva población. Esto permite aumentar mucho el fitness mejor absoluto.

Conclusiones:

En general, no creemos que las gramáticas evolutivas sean muy adecuadas para el caso concreto del multiplexor. No hemos realizado ninguna ejecución que haya dado lugar a los máximos de ninguno de los dos multiplexores (64 y 2048, respectivamente). Pese a que las gramáticas evolutivas suelen funcionar muy bien con un amplio grado de aleatoriedad (en las mutaciones, cruces, etc.) consideramos que el factor más importante para dar lugar a soluciones mejores es, precisamente, el elitismo. Aplicándolo con regeneración da lugar a los mejores resultados. Con elitismo conviene utilizar métodos no tan elitistas (cruce monopunto y selección por ruleta), para no estancar el fitness de la población. También funciona mucho mejor si, además de elitismo, aplicamos un gran porcentaje de mutación y cruce (80% y 90% respectivamente). Las ejecuciones del multiplexor de 2 a 4 son relativamente rápidas pero, en el caso del multiplexor de 3 a 8, el tiempo de ejecución es demasiado alto en comparación con los resultados obtenidos.

Reparto de tareas:

Como todos los métodos de selección, cruce y mutación estaban ya resueltos en otras prácticas anteriores, no ha sido necesario hacer reparto de tareas. El trabajo, tanto con la gramática como con las modificaciones correspondientes a la parte visual y de ejecución, ha sido conjunto en todo momento.