
PRACTICA 3

Objetivo General

Desarrollar la versión preliminar de un Algoritmo Genético para el desarrollo de armaduras planas mediante el Lenguaje C Puro.

Objetivos Particulares:

- Manejo de apuntadores dinámicos en cada una de las partes que conforman el desarrollo de la práctica.
- Aprender a utilizar paso de parámetros por referencia en C.
- Desarrollar un programa desde C, esto es, tomando como base la explicación teórica y plasmarla en un programa funcional.
- Los tres objetivos anteriores se refuerzan con el desarrollo práctico de un Algoritmo Genético que permita simular el desarrollo de armas bidimensionales en C que será base para el resto del curso de Programación Concurrente

Documentos para entregar

La práctica puede desarrollarse en equipo o de forma individual, con equipos de máximo 3 integrantes y los documentos que se necesitan entregar para esta práctica son los siguientes:

1. El programa desarrollado de C para implementar el Algoritmo Genético (40 %).
2. Archivos de prueba con extensión *.dat, para las pruebas preliminares que se realice con el Algoritmo Genético.
3. Informe individual de cómo se abordaron cada uno de los problemas planteados (30% de la calificación). El contenido de este es Caratula, Introducción, Desarrollo y Conclusiones.
4. Exposición exprés centrándose en la implementación, pruebas y resultados y conclusión, con duración máxima de 5 minutos de exposición y 3 minutos de comentarios y preguntas al respecto (30% de la calificación).

Plazo de entrega

La hora y fecha límite para entregarla será el martes 11 de marzo del 2025 durante la sesión del laboratorio, por lo cual la revisión será en vivo dentro de las instalaciones de la UAMI, por lo que no se considera lleva a cabo un examen, de hecho, el resto serán a casa. El programa desarrollado se debe enviar con antelación para el lunes 10 de marzo antes de las 23:59:59 horas.

Nota. No se recibirá ninguna práctica fuera de ese horario, sin ninguna excepción.

Especificaciones del programa para entregar:

Todas las especificaciones de la práctica se darán a conocer el domingo 02 de marzo en el video de apoyo de Semana 3, Sin embargo, se da a conocer las partes fundamentales que tendrá esta Tercera Practica.

Portada: El programa debe contener una portada en C, regresando a los orígenes, esto es, jugando con la terminal y mostrar algo similar a lo que se muestra en la Figura 1:

```

**  **  ****  **  **
**  **  **  **  ***  ***
**  **  ****  **  **  **
*****  **  **  **  **
      ****  **  **  **

PROGRAMACIÓN CONCURRENTE
PRACTICA III
ALGORITMO GENETICO PARA EL DESARROLLO DE ARMADURAS PLANAS
AUTOR.- BENJAMIN MORENO MONTIEL
FECHA DE CREACION: 11/01/2020
Presiona Enter para continuar...|
```

Figura 1. Aspecto visual de la posible portada de la Practica 3.

Menú: Después de la portada se debe implementar un Menú muy sencillo con dos opciones para que el usuario conozca cuales son las opciones del programa, este Menú se muestra en la Figura 2.

```

Practica 2 - Algoritmo Genetico para el desarrollo de Armaduras Planas
Elige una de la siguientes opciones:
1. Abrir un archivo existente e inicializar Algoritmo Genetico
2. Salir del Programa
|
```

Figura 2. Aspecto visual del menú para la Practica 2.

En la Figura 2 podemos observar que se tendrán dos opciones muy sencillas, opción 1 abrir un archivo, opción 2 salir del programa. La opción 1 es la más importante ya que es aquí donde se abre el archivo de prueba con extensión*.dat, el cual debe contener:

1. Probabilidad de cruce, en valor entero.
2. Probabilidad de mutación, en un valor real.
3. Longitud de los cromosomas (60 para esta versión preliminar)
4. Peso total teórico, que para este caso es de 5586.59.
5. Lista teórica de los pesos y la longitud de cada posible elemento en la armadura, el aspecto de estos archivos se muestra en la Figura3.

eje2.c	Menús.c	prueba1.dat	prueba2.da
1	50		
2	12		
3	60		
4	5586.59		
5	1.62	360	
6	1.80	360	
7	1.99	360	
8	2.13	360	
9	2.38	360	
10	2.62	360	
11	5.12	360	
12	7.22	360	
13	2.63	360	

Figura 3. Estructura del archivo que contiene los datos para el Algoritmo Genético.

Modulo para apertura de archivos: Este módulo comprobará si hay archivos existentes, en el caso contrario indicara al usuario que no existió y dará la posibilidad de intentar de nuevo, tal y como se muestra en la Figura 4.

```

APERTURA DE ARCHIVO PARA CARGAR DATOS
Proporciona el nombre del archivo con extension
prueba23.dat
Fichero: prueba23.dat -> Error de Apertura!!! (NO ABIERTO)
Ingresa otro nombre correcto del archivo

APERTURA DE ARCHIVO PARA CARGAR DATOS
Proporciona el nombre del archivo con extension
|

```

Figura 4. Apertura de archivo errónea.

Una vez que un archivo se haya abierto de forma correcta, se desplegarán cada uno de los datos leído y con los cuales se continuará con la ejecución del Algoritmo Genético, esto se muestra en la Figura 5.

```
opelo@Abraxas-Omen2: /mnt/ X + v

APERTURA DE ARCHIVO PARA CARGAR DATOS
Proporciona el nombre del archivo con extension
prueba23.dat
Fichero: prueba23.dat -> Error de Apertura!!! (NO ABIERTO)
Ingresa otro nombre correcto del archivo

APERTURA DE ARCHIVO PARA CARGAR DATOS
Proporciona el nombre del archivo con extension
prueba1.dat
Fichero: prueba1.dat -> Apertura Exitosa!!! (ABIERTO)

Los Datos leidos fueron los siguientes

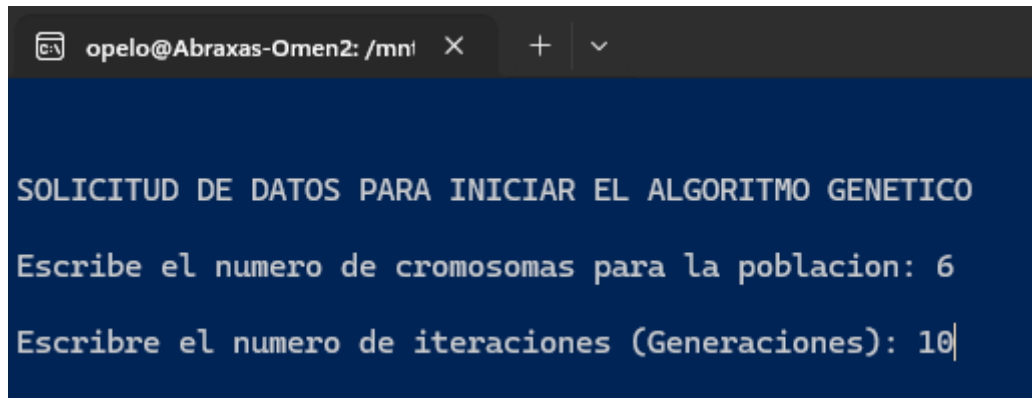
Probabilidad de Cruza = 0.50
Probabilidad de Mutacion = 0.12

Num de objetos = 60
Capacidad maxima = 5586.00

Areas      Longitud
1.62      ,360
1.80      ,360
1.99      ,360
2.13      ,360
2.38      ,360
2.62      ,360
5.12      ,360
7.22      ,360
2.63      ,360
2.88      ,360
4.18      ,360
4.59      ,360
4.80      ,360
4.97      ,360
5.12      ,360
3.84      ,360
3.87      ,360
2.93      ,360
3.09      ,360
3.13      ,360
3.38      ,360
3.47      ,360
16.00     ,360
```

Figura 5. Apertura satisfactoria del archivo de prueba.

Después de abrir el archivo de datos, se pedirá al usuario los datos del tamaño de la población y el número de iteraciones (criterio de paro), tal y como se muestra en la Figura 6.



```

opelo@Abraxas-Omen2: /mn! X + v

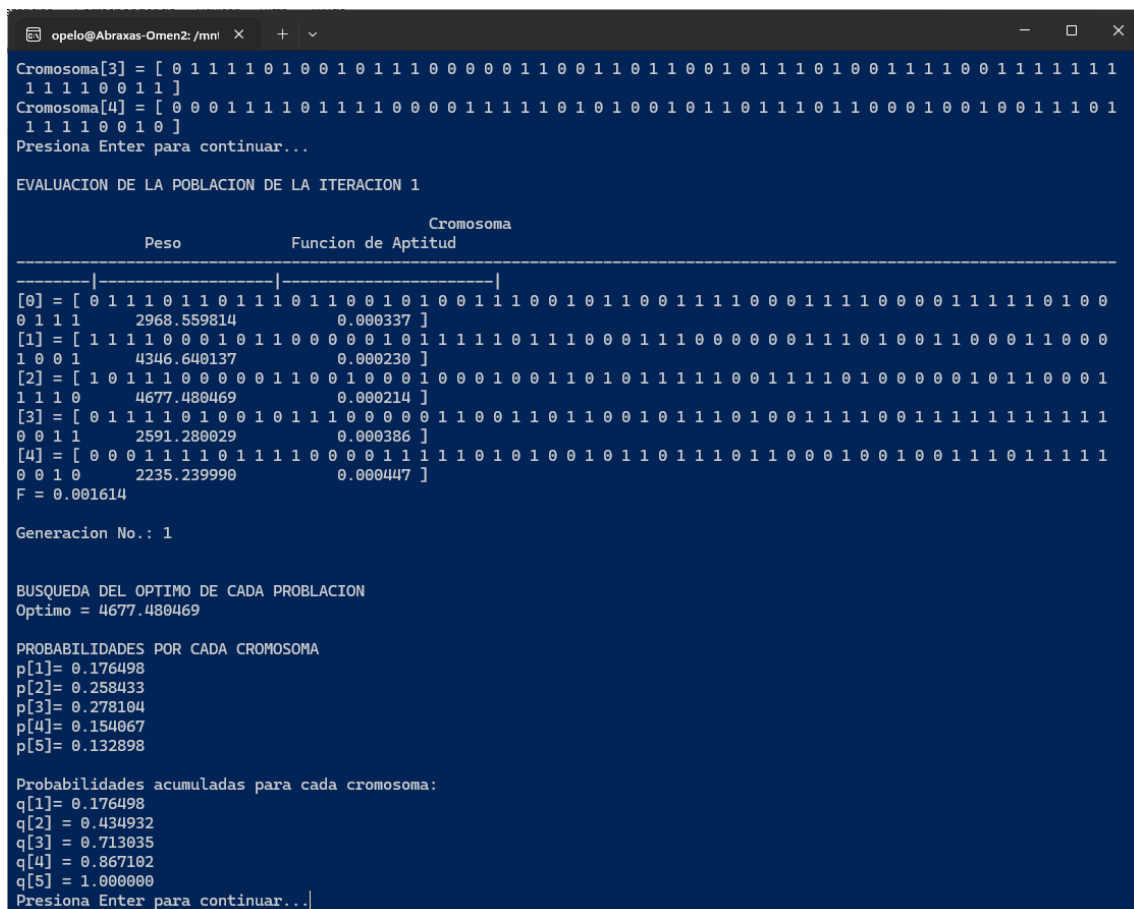
SOLICITUD DE DATOS PARA INICIAR EL ALGORITMO GENETICO

Escribe el numero de cromosomas para la poblacion: 6

Escribe el numero de iteraciones (Generaciones): 10
  
```

Figura 6. Solicitud de los datos para inicial el Algoritmo Genético.

En la Figura 7 Mostramos una posible corrida con el Algoritmo Genético, mostrando cual es la información que se solicita.



```

opelo@Abraxas-Omen2: /mn! X + v

Cromosoma[3] = [ 0 1 1 1 1 0 1 0 0 1 0 1 1 1 0 0 0 0 0 1 1 0 0 1 1 0 1 1 0 0 1 0 1 1 1 0 1 0 0 1 1 1 1 0 0 1 1 1 1 1 1 1
1 1 1 1 0 0 1 1 ]
Cromosoma[4] = [ 0 0 0 1 1 1 1 0 1 1 1 1 0 0 0 0 1 1 1 1 1 0 1 0 1 0 0 1 0 1 1 0 1 1 1 0 1 1 0 0 0 1 0 0 1 0 0 1 1 1 0 1
1 1 1 1 0 0 1 0 ]
Presiona Enter para continuar...

EVALUACION DE LA POBLACION DE LA ITERACION 1

          Cromosoma
      Peso  Funcion de Aptitud
-----|-----|-----|
[0] = [ 0 1 1 1 0 1 1 0 1 1 1 0 1 1 0 0 1 0 1 0 0 1 1 1 0 0 1 0 1 1 0 0 1 1 1 1 0 0 0 1 1 1 1 1 0 1 0 0
0 1 1 1      2968.559814      0.000337 ]
[1] = [ 1 1 1 1 0 0 0 1 0 1 1 0 0 0 0 0 0 1 0 1 1 1 1 1 0 1 1 1 0 0 0 1 1 1 0 0 0 0 0 1 1 1 0 1 0 0 1 1 0 0 0
1 0 0 1      4346.640137      0.000230 ]
[2] = [ 1 0 1 1 1 0 0 0 0 0 1 1 0 0 1 0 0 0 1 0 0 0 1 0 0 1 1 0 1 0 1 1 1 1 0 0 1 1 1 1 0 1 0 0 0 0 0 1 0 1 1 0 0 0 1
1 1 1 0      4677.480469      0.000214 ]
[3] = [ 0 1 1 1 1 0 1 0 0 1 0 1 1 1 0 0 0 0 0 1 1 0 0 1 1 0 1 1 0 0 1 0 1 1 1 0 1 0 0 1 1 1 1 0 0 1 1 1 1 1 1 1 1 1
0 0 1 1      2591.280029      0.000386 ]
[4] = [ 0 0 0 1 1 1 1 0 1 1 1 1 0 0 0 0 1 1 1 1 0 1 0 1 0 0 1 0 1 1 0 1 1 1 0 1 1 0 0 0 1 0 0 1 0 0 1 1 1 0 1 1 1 1 1
0 0 1 0      2235.239990      0.000447 ]
F = 0.001614

Generacion No.: 1

BUSQUEDA DEL OPTIMO DE CADA PROBLACION
Optimo = 4677.480469

PROBABILIDADES POR CADA CROMOSOMA
p[1]= 0.176498
p[2]= 0.258433
p[3]= 0.278104
p[4]= 0.154067
p[5]= 0.132898

Probabilidades acumuladas para cada cromosoma:
q[1]= 0.176498
q[2]= 0.434932
q[3]= 0.713035
q[4]= 0.867102
q[5]= 1.000000
Presiona Enter para continuar...
  
```

```

opelo@Abraxas-Omen2: /mnt/ X + v
METODO PARA REALIZAR LA MUTACION MULTIPUNTO:

probabilidad de mutacion pm = 0.120

numeros aleatorios para mutacion

M'[60]= 0.258000 0.751000 0.621000 0.161000 0.452000 0.608000 0.495000 0.217000 0.834000 0.116000 0.068000 0.850000 0.24
3000 0.913000 0.335000 0.887000 0.613000 0.008000 0.031000 0.545000 0.435000 0.465000 0.128000 0.276000 0.841000 0.25800
0 0.663000 0.090000 0.006000 0.974000 0.257000 0.616000 0.725000 0.229000 0.128000 0.528000 0.188000 0.623000 0.744000 0
.021000 0.090000 0.164000 0.223000 0.332000 0.076000 0.409000 0.570000 0.688000 0.417000 0.952000 0.233000 0.203000 0.41
7000 0.360000 0.478000 0.609000 0.969000 0.141000 0.698000 0.327000
M'[60]= 0.466000 0.306000 0.294000 0.542000 0.886000 0.773000 0.421000 0.073000 0.747000 0.165000 0.446000 0.836000 0.68
0000 0.020000 0.520000 0.755000 0.428000 0.089000 0.442000 0.844000 0.393000 0.026000 0.399000 0.809000 0.738000 0.87600
0 0.417000 0.706000 0.368000 0.467000 0.384000 0.834000 0.124000 0.677000 0.727000 0.010000 0.450000 0.148000 0.434000 0
.196000 0.664000 0.231000 0.384000 0.343000 0.250000 0.903000 0.097000 0.030000 0.343000 0.890000 0.225000 0.735000 0.26
8000 0.623000 0.543000 0.357000 0.499000 0.312000 0.414000 0.866000
M'[60]= 0.136000 0.798000 0.251000 0.253000 0.826000 0.778000 0.614000 0.275000 0.925000 0.400000 0.823000 0.588000 0.63
0000 0.558000 0.930000 0.880000 0.460000 0.378000 0.261000 0.802000 0.619000 0.485000 0.889000 0.886000 0.108000 0.78300
0 0.242000 0.606000 0.094000 0.656000 0.823000 0.223000 0.805000 0.874000 0.828000 0.630000 0.651000 0.793000 0.257000 0
.927000 0.192000 0.079000 0.514000 0.822000 0.636000 0.795000 0.053000 0.447000 0.524000 0.313000 0.248000 0.494000 0.79
7000 0.136000 0.380000 0.904000 0.919000 0.973000 0.861000 0.364000
M'[60]= 0.980000 0.036000 0.939000 0.784000 0.909000 0.118000 0.414000 0.911000 0.910000 0.022000 0.837000 0.102000 0.10
0000 0.702000 0.923000 0.087000 0.848000 0.975000 0.885000 0.723000 0.287000 0.132000 0.216000 0.435000 0.620000 0.59500
0 0.691000 0.890000 0.920000 0.903000 0.605000 0.251000 0.938000 0.543000 0.035000 0.846000 0.660000 0.800000 0.108000 0
.570000 0.821000 0.944000 0.023000 0.272000 0.997000 0.945000 0.710000 0.844000 0.271000 0.594000 0.566000 0.909000 0.07
7000 0.782000 0.343000 0.696000 0.728000 0.033000 0.937000 0.999000
M'[60]= 0.936000 0.894000 0.250000 0.873000 0.436000 0.636000 0.071000 0.448000 0.435000 0.178000 0.369000 0.607000 0.47
4000 0.391000 0.878000 0.470000 0.687000 0.939000 0.666000 0.957000 0.532000 0.583000 0.865000 0.608000 0.716000 0.55900
0 0.656000 0.444000 0.592000 0.944000 0.794000 0.879000 0.837000 0.043000 0.103000 0.625000 0.678000 0.525000 0.424000 0
.464000 0.055000 0.144000 0.422000 0.528000 0.886000 0.651000 0.349000 0.572000 0.589000 0.366000 0.880000 0.120000 0.94
9000 0.096000 0.080000 0.664000 0.654000 0.087000 0.459000 0.245000
Nueva poblacion con mutacion
La poblacion del Algoritmo Genetico es:
Cromosoma[0] = [ 1 1 1 1 0 0 0 1 0 1 1 1 0 0 0 0 0 1 0 1 1 1 1 1 0 1 1 1 0 0 0 1 1 1 0 1 0 0 1 1 0 0 0 0
0 0 0 0 0 0 0 1 ]
Cromosoma[1] = [ 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1
1 0 0 0 1 0 0 0 ]
Cromosoma[2] = [ 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 ]
Cromosoma[3] = [ 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 ]
Cromosoma[4] = [ 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 ]
Cromosoma[5] = [ 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 ]
Presiona Enter para continuar...

EVALUACION DE LA POBLACION DE LA ITERACION 2

Cromosoma
Peso Funcion de Aptitud
-----|-----|-----|
[0] = [ 1 1 1 1 0 0 0 1 0 1 1 1 0 0 0 0 0 1 0 1 1 1 1 1 0 1 1 1 0 0 0 1 1 1 0 1 0 0 1 1 0 0 0 0 0 0 0 0
0 0 0 1 4181.759766 0.000239 ]
[1] = [ 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0
1 0 0 0 772.920044 0.001294 ]
[2] = [ 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 1
1 1 1 0 1629.719727 0.000614 ]
[3] = [ 1 0 1 1 1 0 0 0 0 0 1 1 0 0 0 1 0 0 0 1 0 0 0 1 1 0 1 0 1 1 1 1 1 0 1 1 1 0 1 0 0 0 0 0 1 0 0 0 0 0 0 0
0 0 0 0 3741.479736 0.000267 ]
[4] = [ 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 583.200012 0.001715 ]
F = 0.004128

Generacion No.: 2

BUSQUEDA DEL OPTIMO DE CADA PROBLACION
Optimo = 4677.480469

PROBABILIDADES POR CADA CROMOSOMA
p[1]= 0.383328
p[2]= 0.070851
p[3]= 0.149391
p[4]= 0.342969
p[5]= 0.053460

Probabilidades acumuladas para cada cromosoma:
q[1]= 0.383328
q[2]= 0.454179
q[3]= 0.603571
q[4]= 0.946540
q[5]= 1.000000
Presiona Enter para continuar...

```

Figura 7. Ejecución para la primera iteración del genético.

Finalmente mostramos la salida del algoritmo genético después de las 10 iteraciones que fueron fijadas al inicio del programa, lo cual se muestra en la Figura 8.

```
EVALUACION DE LA POBLACION DE LA ITERACION 10
```

	Peso	Funcion de Aptitud	Cromosoma
[0] = [1 1 1 1 0 1 0 1 0 1 0 0	1 0 0 0 0 1 0 1 0 1 0 0 0 0 0 0 0 0 0 1 0 1 0 0 1 1 1 1 0 1 1 1 0 0 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 1 1900.079956 0.000526]
[1] = [1 1 1 0 0 1 0 1 0 1 0 0	1 0 0 0 0 1 0 1 0 1 0 0 0 0 0 0 0 0 0 1 0 1 0 0 1 1 1 1 0 1 1 1 0 0 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 1 2336.399902 0.000428]
[2] = [1 1 1 1 0 0 0 1 0 0 1 0 0	1 0 0 0 0 1 0 1 0 1 0 1 0 0 0 0 0 1 0 0 1 0 0 0 1 1 1 1 0 1 1 1 0 0 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 3276.359863 0.000305]
[3] = [1 1 1 1 0 0 0 1 0 0 1 0 0	0 1 0 0 0 1 0 1 0 1 0 0 0 0 0 0 1 0 0 1 0 0 0 1 1 1 1 0 1 1 0 0 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 3276.359863 0.000305]
[4] = [1 0 1 1 0 0 0 1 0 0 1 0 0	0 1 0 1 0 1 0 1 0 0 0 0 0 1 0 0 1 0 0 0 1 1 1 1 0 1 1 0 0 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 2725.199951 0.000367]
[5] = [1 0 1 1 0 1 0 1 0 1 0 0	1 0 0 0 1 0 1 0 1 0 1 0 0 0 0 0 0 0 0 1 0 1 0 0 1 1 1 0 1 1 0 0 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 1 1933.199951 0.000517]

F = 0.002449

Generacion No.: 10

BUSQUEDA DEL OPTIMO DE CADA POBLACION

Optimo = 3804.839844

Resultado obtenido en la iteracion No.: 0

Peso mas Optimo de: 3804.84

Cromosoma optimo= 1 0 1 1 1 1 1 1 0 1 0 0 1 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1 1 0 0 1 1 1 1 1 1 0 1 0 0 1 1 0 1 1 1 1 0 1 1 1 0 1 0 1

Presiona Enter para continuar...

Figura 8. Resultados finales del algoritmo genético.