

Tarea 8 - Optimización estocástica

Erick Salvador Alvarez Valencia

CIMAT A.C.,
`erick.alvarez@cimat.mx`

Resumen En el presente reporte se hablará sobre las correcciones realizadas al algoritmo genético implementado en la tarea anterior, además se incluirá un nuevo conjunto de resultados obtenidos de aplicar la nueva versión del algoritmo.

A parte de lo anterior también se hablará sobre los ejercicios extras mencionados en la descripción de la tarea, se incluirán los resultados obtenidos y se darán algunas conclusiones.

Keywords: Algoritmo genético, cruce, mutación, mapeo de fenotipos.

1. Reimplementación del algoritmo genético

Para esta tarea lo que se hizo fue hacer las correcciones necesarias a la versión del algoritmo genético implementado en la tarea anterior. A continuación se muestra la lista de correcciones hechas al algoritmo:

1. **Corrección al número de candidatos en el torneo** : En esta versión del algoritmo se implementó la versión correcta del torneo binario el cual toma dos individuos aleatoriamente de la población y regresa el que mejor aptitud ofrezca. En cambio con la versión anterior donde se tomaban K individuos para hacer el torneo, esto generaba bastante presión de selección, lo cual no era bueno.
2. **Corrección al número de individuos que sobrevivían** : En la versión anterior del algoritmo se tenía programado que para la nueva población sobrevivía una cantidad de M individuos donde el elitismo solamente se mantiene en un individuo, esto para evitar convergencia prematura. Además se programó que el mejor individuo de la población anterior solamente sobrevivía si no había ningún individuo en la nueva población que lo supere, en caso de sobrevivir este reemplaza a algún individuo aleatorio de la nueva población.
3. **Modificación de la cruce** : En la versión anterior del algoritmo se tenía programada la cruce de un punto en donde el corte se hacía uno por individuo, y en esta ocasión se hace uno por variable. Además se implementó de la misma forma la cruce de dos puntos. Más adelante se describirá cómo se hizo.

En general estas fueron las correcciones principales que se hicieron al algoritmo genético, todo lo demás estaba de acuerdo a lo establecido. En la siguiente sección se mostrará las pruebas realizadas del algoritmo, pero antes se describirá la implementación de la cruce a dos puntos en el algoritmo genetico, ya que esta se usó para hacer las pruebas. Para esta cruce solamente se toma la variable y se generan dos puntos de corte de manera aleatoria, por lo cual tendremos tres pedazos de esta variable los cuales los heredamos a los hijos de forma secuencial. A continuación se incluye el pseudocódigo de dicho operador de cruce.

Algorithm 1 Cruza de dos puntos.

```

1: procedure 2PCROSSOVER( $p1, p2$ )
2:    $d \leftarrow \text{size}(p1)$ .
3:    $\text{son1} \leftarrow \{\}$ .
4:    $\text{son2} \leftarrow \{\}$ .
5:   for Cada Variable  $v1, v2$  de  $p1, p2$  do
6:      $\text{left} \leftarrow 0$ .
7:      $\text{right} \leftarrow 0$ .
8:     while  $\text{left} = \text{right}$  do
9:        $\text{left} \leftarrow \text{rand}(2, \text{size}(v1) - 1)$ .
10:       $\text{right} \leftarrow \text{rand}(2, \text{size}(v1) - 1)$ .     $\triangleright$  Calculamos los puntos de corte de la
variable, excluyendo las orillas.
11:       $v1\text{New} \leftarrow v1[1:\text{left}] \cup v2[\text{left} + 1:\text{right}] \cup v1[\text{right} + 1:\text{size}(v1)]$ .
12:       $v2\text{New} \leftarrow v2[1:\text{left}] \cup v1[\text{left} + 1:\text{right}] \cup v2[\text{right} + 1:\text{size}(v1)]$ .
13:       $\text{son1} \leftarrow \text{son1} \cup v1\text{New}$ .
14:       $\text{son2} \leftarrow \text{son2} \cup v2\text{New}$ .
15:   return  $\text{son1}, \text{son2}$ .
```

2. Ejecuciones y resultados del AG

En esta parte se mostrarán los resultados obtenidos al implementar el algoritmo genético para minimizar un conjunto de funciones, en general se usaron las mismas funciones que en la tarea anterior, así como la representación de los individuos y el mapeo a los reales. A continuación se muestra el conjunto de funciones trabajado.

1. **Sphere** : $\sum_{i=1}^d x_i^2$
2. **Ellipsoid** : $\sum_{i=1}^d 10^{6 \frac{i-1}{d-1}} x_i^2$
3. **Zakharov** : $\sum_{i=1}^d x_i^2 + (\sum_{i=1}^d 0,5ix_i^2)^2 + (\sum_{i=1}^d 0,5ix_i^2)^4$
4. **Rosenbrock** : $\sum_{i=1}^{d-1} [(1 - x_i)^2 + 100(x_{i+1} - x_i^2)^2]$
5. **Ackley** : $-20 \exp(-0,2 \sqrt{\frac{1}{d} \sum_{i=1}^d x_i^2}) - \exp(\frac{1}{d} \sum_{i=1}^d \cos(2\pi x_i)) + 20 + e$
6. **Griewangk** : $\sum_{i=1}^d \frac{x_i^2}{4000} - \prod_{i=1}^d \cos(\frac{x_i}{\sqrt{i}}) + 1$
7. **Rastrigin** : $10d + \sum_{i=1}^d [x_i^2 - 10\cos(2\pi x_i)]$

Para cada una de estas funciones se hicieron pruebas con el algoritmo genético usando la cruza de uno y dos puntos de corte. Lo que si cambió con respecto a la tarea anterior fueron algunos de los parámetros de ejecución, para este caso se redujo el número máximo de evaluaciones de la función a 150000 y la probabilidad de mutación se fijó en $\frac{1}{18}$. Además de los parámetros anteriores se hicieron pruebas con diferentes tamaños de población pero con la que mejor resultados dió fue con 100 individuos.

A continuación se muestran los resultados obtenidos de realizar 50 ejecuciones del AG con los parámetros anteriormente descritos, y donde se usó cruza de un punto. En dichos resultados se muestran algunos valores estadísticos con respecto al error (fitness obtenido al final de la ejecución) tales como media, mediana, desviación estándar, entre otros.

Cuadro 1: Resultados del AG con cruza de un punto

Función	Promedio de error	Desviación estándar del error	Mediana del error	Mejor fitness	Peor fitness
Sphere	1172.75	575.115	1039.14	290.598	2420.77
Ellipsoid	0.00115668	0.000425307	0.0011033	0.000457571	0.00264921
Zakharov	29.7211	17.6809	25.7927	3.22426	95.1794
Rosenbrock	508.718	462.728	340.146	67.0409	2318.21
Ackley	2.9193	0.454573	2.88085	1.87144	4.23727
Griewangk	1.32363	0.169788	1.275	1.06989	1.90615
Rastrigin	42.836	11.8337	42.9332	18.7197	70.3445

En la tabla anterior se aprecian los resultados obtenidos en las 50 ejecuciones del algoritmo para todas las funciones, en general podemos ver que no son tan buenos, ya que en promedio el error fue bastante alto, además en general el mejor fitness siempre estaba en valores alrededor de 11-17. Otro parámetro que nos confirma el hecho de que en la mayoría de las ocasiones el algoritmo tenía errores altos fue la desviación estándar del error, ya que ésta en la mayoría de las funciones fue muy baja, lo cual nos indica que el error generalmente se mantuvo en valores altos.

Otra cosa que se puede notar es que la función para la que peores resultados se obtuvo fue la esfera, lo cual es muy inusual ya que esta función es en general muy fácil de optimizar ya que no cuenta con muchas dependencias en las variables. De igual manera hubieron algunas funciones que en general si se optimizaron bien, como la Elipsoide y la Griewangk.

Podemos dar una pequeña conclusión de estas ejecuciones. El hecho de que los resultados obtenidos por el algoritmo no fueran tan buenos es porque el valor de mutación usado es en general bastante alto, por lo cual se produce mucha diversidad en la población, esto se confirmará en la siguiente sección. Además de que el número de evaluaciones de la función no fue el suficiente para obtener mejores resultados.

Ahora se mostrarán los resultados del AG con cruza de dos puntos.

Cuadro 2: Resultados del AG con cruza de dos puntos

Función	Promedio de error	Desviación estándar del error	Mediana del error	Mejor fitness	Peor fitness
Sphere	3285.85	1414.53	3236.62	505.926	6223.51
Ellipsoid	0.00234562	0.00109643	0.00214098	0.00074875	0.005534
Zakharov	44.3668	22.2282	40.0207	12.3532	112.844
Rosenbrock	1186.09	761.588	956.749	85.9205	3535.26
Ackley	3.80242	0.474602	3.79624	2.86839	4.90875
Griewangk	1.88715	0.385154	1.87028	1.25165	2.87504
Rastrigin	42.836	11.8337	42.9332	18.7197	70.3445

En la Tabla anterior podemos ver los resultados obtenidos al aplicar cruza de dos puntos en el algoritmo, lo primero que podemos notar es que en general se obtuvieron peores resultados que con la cruza de dos puntos, y esto se nota claramente en la función de la Esfera. Esto por lo que ya se mencionó anteriormente, la mutación alta provoca mucha diversidad entre los individuos y dificulta al algoritmo el llegar a una convergencia. Además de que el cruce de dos puntos genera un poco más de diversidad en los hijos ya que estos no serán tan parecidos a los padres como lo es en la cruza de un punto, esto explica el porque se empeoraron los resultados obtenidos con la cruza de un punto.

Para demostrar el efecto del ratio de mutación alto, se hizo otro conjunto de ejecuciones al algoritmo, usando los mismos parámetros y la cruza de dos puntos, pero esta vez se utilizó una probabilidad de mutación más baja ($\frac{1}{180}$). A continuación se muestran los resultados obtenidos.

Cuadro 3: Resultados del AG con cruza de dos puntos y prob. de mutación baja

Función	Promedio de error	Desviación estándar del error	Mediana del error	Mejor fitness	Peor fitness
Sphere	0.000863762	0.000118171	0.000890584	0.000513395	0.000974404
Ellipsoid	0.000862938	0.000148274	0.00092226	0.000200494	0.000989659
Zakharov	54.0387	56.1087	35.4809	0.00032145	175.891
Rosenbrock	138.944	472.841	8.96611	7.49521	3316.71
Ackley	0.0008866	0.000123591	0.000918321	0.000563772	0.00099687
Griewangk	0.0395792	0.0353509	0.0322902	0.000578868	0.173034
Rastrigin	42.836	11.8337	42.9332	18.7197	70.3445

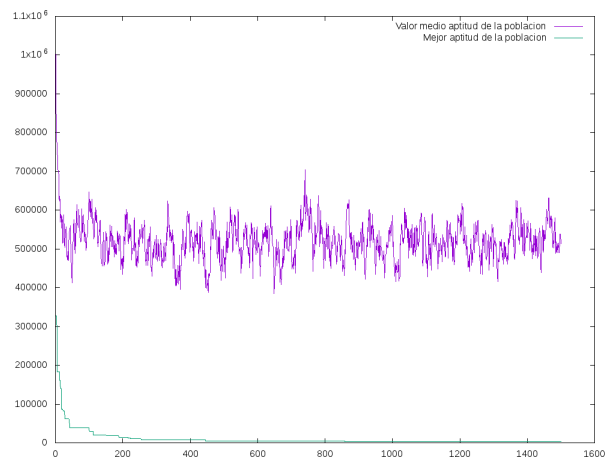
Como podemos ver en esta tercer Tabla los resultados obtenidos fueron mucho mejores a los de las dos Tablas anteriores, y como se comentó, solamente se modificó el ratio de mutación. Se puede apreciar que hay funciones para las cuales en el mejor de los casos se encontró un fitness muy bajo, con magnitud de 10^{-3} , y tal como en el caso anterior, las funciones que le costó más optimizar fueron la 3, 4 y la 7.

3. Evolución de la aptitud en la Esfera

En esta parte simplemente se mostrarán dos gráficas las cuales corresponden a la evolución del mejor valor de aptitud y el valor medio en aptitud en la

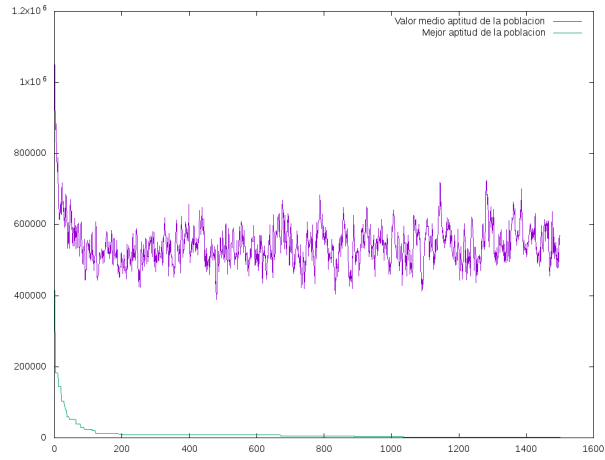
población que va tomando el AG a lo largo de una ejecución en la evolución de sus poblaciones.

Estas gráficas fueron obtenidas en una corrida del algoritmo usando la función de la Esfera para optimizar. A continuación se muestra la gráfica de la evolución del mejor fitness y el fitness medio de la función de la Esfera usando cruza de un punto.



(a) Figura 1. Evolución del mejor fitness y el fitness medio para la función Sphere con cruza de un punto.

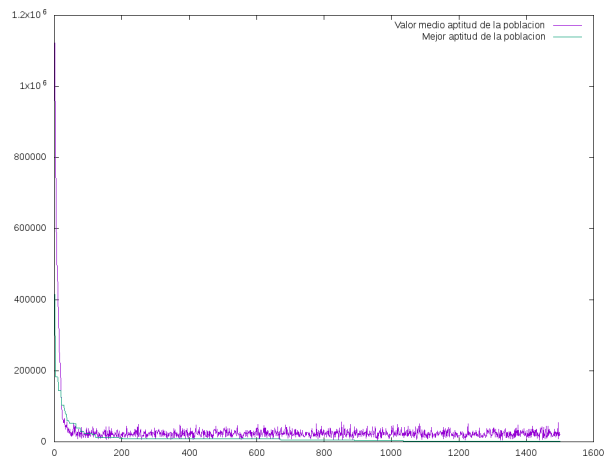
Como podemos ver en la Figura anterior, la evolución del fitness medio (azul) está bastante separado a la evolución del mejor fitness (amarillo), tal que el primero vive en el orden de los cientos de miles de unidades mientras que el segundo vive en el orden de las centenas, además de que el fitness medio oscila en una forma descomunal. Esto es por lo que se explicaba en la sección anterior, en general nuestra población tiene bastante diversidad y si la analizáramos, veríamos que la gran mayoría de sus individuos produce un fitness bastante alto, mientras que unos pocos son los que tienen fitness bajos, y esto es gracias al elitismo que estamos manteniendo en cada nueva generación de individuos. Ahora se mostrará la una gráfica relacionando los dos valores anteriores solamente que ahora se utilizó cruza de dos puntos en el algoritmo.



(b) Figura 2. Evolución de l mejor fitness y el fitness medio para la función Sphere con cruza de dos puntos.

En la Figura 2 podemos apreciar un comportamiento similar en el fitness medio de la población al mostrado en la Figura 1, y esto coincide con los resultados mostrados en la sección anterior, donde en general se obtuvieron peores resultados en la cruza de dos puntos, pero de igual manera vemos que hay mucha diversidad en la población.

Por último se mostrará una gráfica en donde se hace cruza de un punto pero se coloca un valor para el ratio de mutación bajo, el mismo que se usó en las pruebas de la sección pasada ($\frac{1}{180}$).



(c) Figura 3. Evolución de l mejor fitness y el fitness medio para la función Sphere con cruza de un punto y valor de mutación bajo.

Para esta tercer gráfica vemos una diferencia notable en el comportamiento de la evolución del fitness medio en donde se aprecia que este es mucho más bajo, vemos que está casi pegado al mejor fitness de la población, además de que la magnitud de su oscilación es más baja.

4. Análisis del conjunto de hijos generados por dos padres arbitrarios

En esta sección se analizarán los posibles hijos que son generados por una elección arbitraria de dos padres usando la cruza de uno y dos puntos. En general se tomaron 3 conjuntos de padres y se aplicó los dos tipos de cruza, se generaron todos los posibles hijos con esa cruza y se graficaron. Para poder hacer la visualización se modificó la dimensión de los individuos a 2 dimensiones. A continuación se mostrarán las primeras dos gráficas obtenidas por la primer selección de padres.

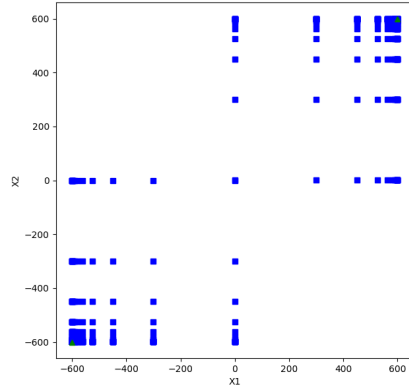


Figura 1: Conjunto de hijos generados por cruce de un punto

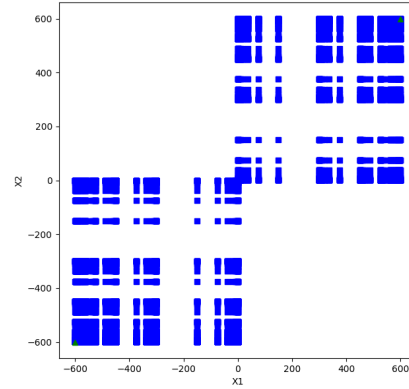


Figura 2: Conjunto de hijos generados por cruce de dos puntos

En las dos gráficas anteriores se muestran al conjunto de hijos generados por las cruces de uno y dos puntos, los cuales se muestran como cuadrados azules. Y de la misma forma se muestran a los padres que generaron a dichos hijos, como triángulos verdes.

Para este primer caso se tomaron los padres en las posiciones $(-600, -600)$ y $(600, 600)$ lo que corresponde a poner todos los bits en 0 y en 1 respectivamente, ya que el dominio de la función de la Esfera es en el rango $(-600, 600)$. Se puede notar que para esta selección, el conjunto de hijos tendió a llenar el primer y tercer cuadrantes de la gráfica, esto se ve más claro en el conjunto de hijos generados por cruce de dos puntos, lo cual nos podría sugerir que para una selección de padres de este estilo los hijos que se generan lo hacen dentro de una pequeña área que hay entre los dos padres y no regados por el espacio de manera aleatoria como se podría pensar.

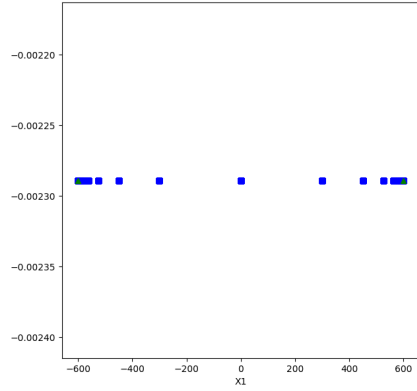


Figura 3: Conjunto de hijos generados por cruza de un punto

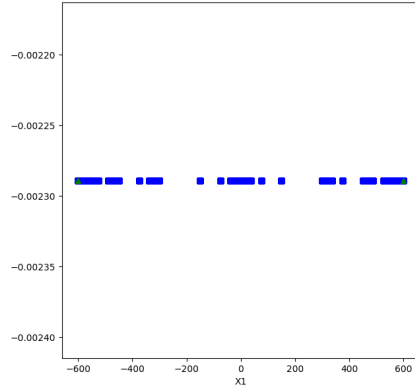


Figura 4: Conjunto de hijos generados por cruza de dos puntos

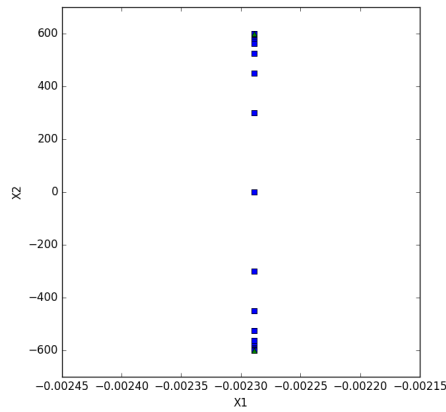


Figura 5: Conjunto de hijos generados por cruza de un punto

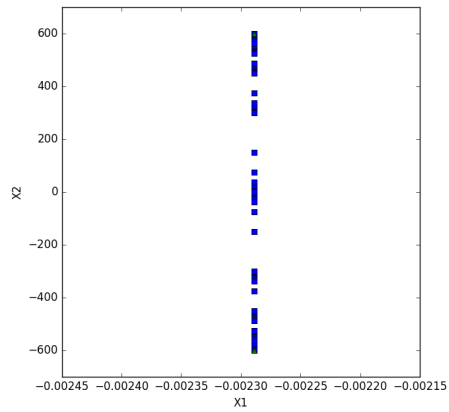


Figura 6: Conjunto de hijos generados por cruza de dos puntos

Para estas 4 gráficas anteriores la selección de padres se hizo dentro de los ejes X y Y, en general las coordenadas fueron (-600, 0) y (600, 0) para el primer conjunto, (0, 600) y (0, -600) para el segundo conjunto. Para representar el cero en la representación binaria se usaron $\frac{2^{18}-1}{2}$ bits, lo cual correspondería a tomar el conjunto de todos los bits prendidos y recorrerlo en una unidad hacia la derecha

llenando el espacio de más a la izquierda con un cero.

Podemos ver que para estos dos casos, los hijos tienden a generarse en el eje donde se tomaron los padres, además estos tienen un poco más tendencia a generarse en los extremos.

Otro caso interesante sería cuando los padres son iguales, en dicha situación tendríamos que todos los hijos serían iguales a los padres, y esto tiene sentido ya que los componentes genéticos de los padres son iguales.

5. Ejercicio sobre la presión de selección

Para este último ejercicio se implementó el algoritmo pedido en la descripción de la tarea, en donde se busca ver el efecto que tiene la selección por torneo binario en contra de la selección por ruleta. Para este caso se aplicaron ambas en un problema de maximización donde tenemos una población de N individuos y para las nuevas poblaciones generamos a la descendencia con las funciones de selección por torneo binario o selección por ruleta. La idea es hacer esto hasta que se logre que toda la población converga a un cierto valor.

Este ejercicio se realizó con la misma población inicial para ambos tipos de selección, además se hicieron 100 ejecuciones en las cuales se cambió la semilla de números aleatorios y se aplicó la evolución de la población para ambos tipos de selección. Para cada iteración se mostró el número de iteraciones que tardó cada selección para converger a cierto valor, además de que se imprimió dicho valor. Al final se imprimió el promedio de iteraciones en las que tardó en converger cada algoritmo. A continuación se muestran los resultados obtenidos por el programa.

```
estocástica/Tarea 8/Code/Presion$ ./main
Iteración 1
Selección por torneo. N = 7, Valor = 10
Ruleta. N = 19, Valor = 9

Iteración 2
Selección por torneo. N = 7, Valor = 10
Ruleta. N = 20, Valor = 10

Iteración 3
Selección por torneo. N = 7, Valor = 10
Ruleta. N = 18, Valor = 10

Iteración 4
Selección por torneo. N = 9, Valor = 10
Ruleta. N = 58, Valor = 10

Iteración 5
Selección por torneo. N = 8, Valor = 10
Ruleta. N = 68, Valor = 10

Iteración 6
Selección por torneo. N = 9, Valor = 10
Ruleta. N = 42, Valor = 10

Iteración 7
Selección por torneo. N = 9, Valor = 10
Ruleta. N = 39, Valor = 10

Iteración 8
Selección por torneo. N = 7, Valor = 10
Ruleta. N = 73, Valor = 10

Iteración 9
Selección por torneo. N = 6, Valor = 10
Ruleta. N = 31, Valor = 9

Iteración 10
Selección por torneo. N = 7, Valor = 10
Ruleta. N = 25, Valor = 10

Iteración 11
Selección por torneo. N = 7, Valor = 10
Ruleta. N = 66, Valor = 10

Iteración 12
Selección por torneo. N = 7, Valor = 10
Ruleta. N = 45, Valor = 10

Iteración 13
Selección por torneo. N = 6, Valor = 10
Ruleta. N = 47, Valor = 10

Iteración 14
Selección por torneo. N = 6, Valor = 10
Ruleta. N = 47, Valor = 10
```

(a) Figura 4. Resultado de las iteraciones del algoritmo, se muestra el valor al que convergió la población y el número de iteraciones que tardó el algoritmo para converger.

```

Iteración 94
Selección por torneo. N = 7, Valor = 10
Ruleta. N = 38, Valor = 10

Iteración 95
Selección por torneo. N = 7, Valor = 10
Ruleta. N = 37, Valor = 10

Iteración 96
Selección por torneo. N = 7, Valor = 10
Ruleta. N = 20, Valor = 10

Iteración 97
Selección por torneo. N = 7, Valor = 10
Ruleta. N = 45, Valor = 10

Iteración 98
Selección por torneo. N = 9, Valor = 10
Ruleta. N = 19, Valor = 10

Iteración 99
Selección por torneo. N = 7, Valor = 10
Ruleta. N = 23, Valor = 10

Iteración 100
Selección por torneo. N = 7, Valor = 10
Ruleta. N = 36, Valor = 10

Terminado.
Promedio N selección por torneo: 7.35
Promedio N ruleta: 33.22

```

(b) Figura 5. Resultado de las 100 iteraciones del algoritmo, se muestra el promedio de iteraciones que tardó el mismo para converger.

En las dos imágenes anteriores se muestran los resultados del programa. En la primer imagen se muestra el resultado de algunas iteraciones del algoritmo, podemos notar que la selección por torneo tiende a converger la población al valor máximo el cual es 10. Para el caso de la ruleta también hay cierta tendencia a converger la población al valor máximo, pero de la misma forma hay pocos casos en donde la población converge a valores distintos, como es el 9 o el 8. Además se puede notar que en general la selección por torneo tiende a converger más rápido que la selección por ruleta, y esto lo confirmamos en la Figura 5. donde el promedio de la selección por torneo fue de 7.35 y la selección por ruleta fue de 33.22.

Finalmente podemos concluir que la selección por torneo binario tiene más presión de selección que la de ruleta, incluso teniendo en cuenta que el torneo solamente contiene 2 candidatos, en el caso de que tomáramos más individuos habría mucha más presión de selección.

5.1. Comentarios finales

En el presente reporte se describió la reimplementación del algoritmo genético hecho en la tarea pasada, y se aplicó al mismo conjunto de funciones pero se usaron distintos parámetros. Se vio que los resultados del algoritmo no fueron tan buenos con respecto a la minimización de las funciones, esto debido a que la probabilidad de mutación no fue tan pequeña, y se demostró que poniendo una probabilidad más pequeña se obtuvieron mejores resultados.

Además se implementó un generador de hijos con cruce de uno y dos puntos, y se obtuvieron resultados interesantes de acuerdo a la selección de dichos padres,

en general se pudo ver que la cruza de dos puntos tiende a generar hijos en las mismas áreas que la cruza de un punto pero la de dos puntos intensifica dicha generación de hijos, es decir, genera hijos en la misma área que la cruza de un punto pero en ciertas zonas en las que la cruza de un punto no las generaría. Finalmente se realizó un ejercicio sobre la presión de selección aplicando dos métodos de selección comúnmente usados por el algoritmo genético y se pudo apreciar que el método de selección por torneo tiende a tener más presión en comparación a la selección por ruleta. Además de que la selección por torneo maximizó mejor en comparación al otro método de selección, esto viene implícito en la presión de selección.