



Universidad Autónoma de Querétaro  
Facultad De Ingeniería



**FACULTAD DE  
INGENIERÍA**

# **Algoritmo Genetico Dicotomico**

## **Algoritmos Metaheurísticos Tarea 2**

P R E S E N T A

**Ing. Daniel Eduardo González Alvarado**

Profesor

**Dr. Marco Antonio Aceves Fernández**

12 de Septiembre de 2025

# Índice

<b>1. Introducción</b>	<b>1</b>
1.1. Conceptos Básicos . . . . .	1
1.2. Componentes de los Algoritmos Genéticos . . . . .	1
1.3. Aplicacion a Algoritmo Dicotómico . . . . .	3
<b>2. Desarrollo</b>	<b>4</b>
2.1. Generación de la Población Inicial . . . . .	4
2.2. Selección . . . . .	4
2.2.1. Rank . . . . .	5
2.2.2. Torneo . . . . .	6
2.2.3. Seleccin de individuos para la siguiente generación . . . . .	6
2.3. Cruzamiento . . . . .	7
2.4. Analisis del Algoritmo Genético . . . . .	7
2.5. Segundo Metodo de cruce . . . . .	9
<b>3. Resultados</b>	<b>10</b>
<b>4. Discusión</b>	<b>11</b>
<b>5. Conclusión</b>	<b>11</b>
<b>6. Bibliografía</b>	<b>12</b>

# 1. Introducción

Los algoritmos genéticos (AG) son una técnica de optimización inspirada en los principios de la evolución natural y la genética basados en la mecánica de selección natural. Combinando la supervivencia del más apto con la reproducción sexual, con intercambio de información estructurada [1].

En la naturaleza, los organismos más aptos tienen una mayor probabilidad de sobrevivir y reproducirse, transmitiendo sus genes a la siguiente generación. De manera similar, en los AG, las soluciones potenciales (individuos) se representan como cadenas de caracteres (cromosomas) y se evalúan mediante una función de aptitud que mide su calidad. De tal manera que la combinación de buenos ancestros puede generar descendencia aún mejor.

Estos algoritmos se utilizan para resolver problemas complejos mediante la simulación de procesos evolutivos, como la selección, el cruce y la mutación. Los AG son especialmente útiles en problemas donde el espacio de búsqueda es grande y no se dispone de una solución analítica directa.

## 1.1. Conceptos Básicos

- **Población:** Conjunto de soluciones candidatas que evolucionan a lo largo del tiempo.
- **Generación:** Cada iteración del algoritmo, donde se aplican los operadores genéticos.
- **Cromosoma:** Representación de una solución en la población, comúnmente como una cadena de bits.
- **Alelo:** Valor específico que puede tomar un gen en un cromosoma.
- **Gen:** Elemento básico de un cromosoma, que representa una característica de la solución.

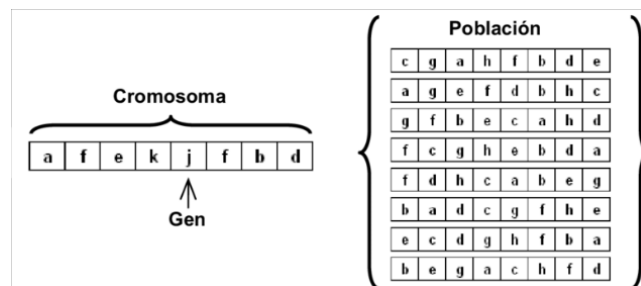


Figura 1: Representación de Poblacion y Cromosoma

## 1.2. Componentes de los Algoritmos Genéticos

- **Codificación de individuos:** Representación de las posibles soluciones del problema, comúnmente en forma de cadenas binarias, vectores o estructuras más complejas.
- **Función de aptitud (fitness):** Mide la calidad de cada individuo dentro de la población, es decir, qué tan buena es una solución respecto al objetivo del problema.

- **Población inicial:** Conjunto de individuos generados aleatoriamente o a partir de heurísticas, que servirán como punto de partida para la evolución.
- **Operadores genéticos:**
  - **Selección:** Proceso mediante el cual se eligen los individuos que participarán en la reproducción, favoreciendo a los de mayor aptitud.
    - **Selección por torneo:** Se seleccionan aleatoriamente varios individuos y se elige el mejor entre ellos.
    - **Selección por ruleta:** Cada individuo tiene una probabilidad de ser seleccionado proporcional a su aptitud.
    - **Selección por rango:** Los individuos se ordenan según su aptitud y se asignan probabilidades de selección basadas en su posición en el ranking.
  - **Cruzamiento (crossover):** Combina el material genético de dos padres para generar descendencia con características de ambos.
    - **Cruzamiento de un punto:** Se selecciona un punto de corte en los cromosomas de los padres y se intercambian las partes posteriores a ese punto.
    - **Cruzamiento de k puntos:** Se seleccionan k puntos de corte y se intercambian las secciones entre esos puntos.
    - **Cruzamiento aritmético:** Se combinan los valores de los padres mediante una operación aritmética (por ejemplo, promedio).
    - **Cruzamiento de orden:** Utilizado en problemas de permutación, donde se preserva el orden relativo de los genes.
    - **Cruzamiento de ciclo:** También para problemas de permutación, donde se identifican ciclos en los cromosomas de los padres para crear hijos.
    - **Cruzamiento basado en la posición:** Se seleccionan posiciones específicas en los cromosomas de los padres para formar el hijo.
    - **Cruzamiento de mezcla:** Se combinan segmentos de los padres de manera aleatoria.
  - **Mutación:** Introduce variaciones aleatorias en los individuos para mantener la diversidad genética y evitar la convergencia prematura.
- **Criterios de paro:** Condiciones que determinan el final de la ejecución del algoritmo, como alcanzar un número máximo de generaciones o una solución suficientemente buena.

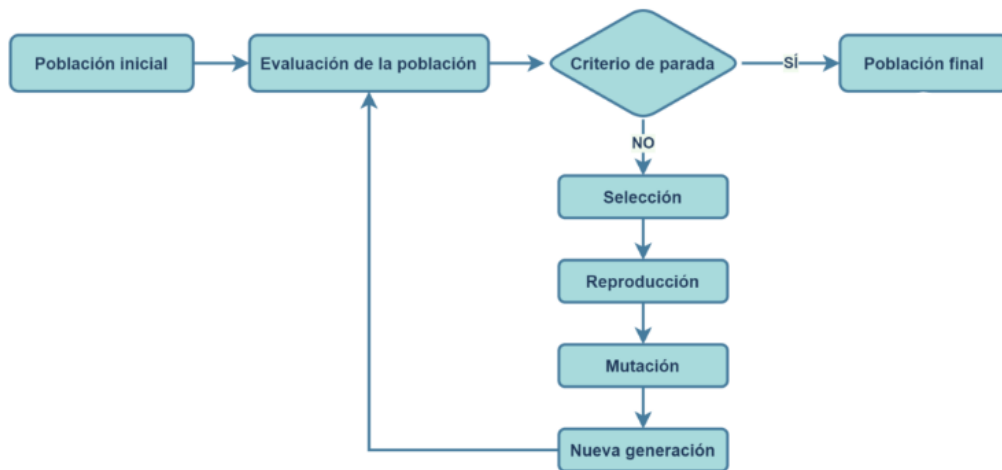


Figura 2: Ciclo de un Algoritmo Genético

### 1.3. Aplicacion a Algoritmo Dicotómico

Para nuestro problema, desarrollaremos un AG que busque los mejores individuos según nuestra función de aptitud, la cual es  $f(x) = x^2$ . Por lo tanto tomaremos en cuenta los siguientes aspectos:

- Aptitud entre 0 y 1.
- Cromosoma: Binario de 6 bits.
- Alelos: 0 y 1.
- Función de aptitud:  $f(x) = x^2$ .
- Espacio de búsqueda:  $\{-1, 1\}[0, 63]$ .

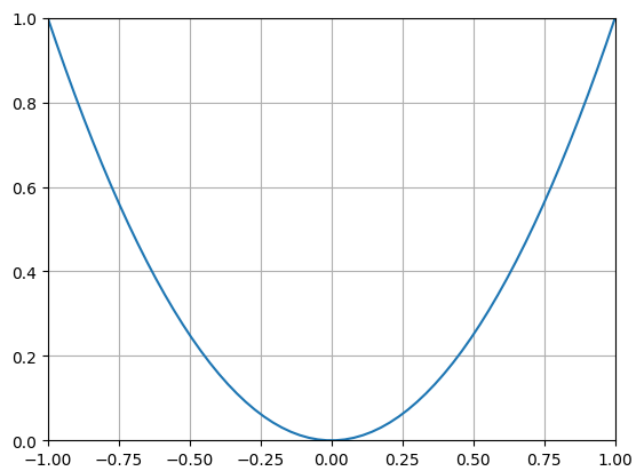


Figura 3: Funcion de Aptitud  $f(x) = x^2$

## 2. Desarrollo

### 2.1. Generación de la Población Inicial

La población inicial se generará aleatoriamente, buscando que ningún individuo tenga una aptitud mayor al 0.9, para poder tener una mejor apreciación del cambio según la generación, creando un conjunto de cromosomas binarios de 6 bits. Cada cromosoma representará un individuo en la población y se generará de la siguiente manera:

- Se crearán 10 individuos (cromosomas) en la población inicial.
- Cada cromosoma será una cadena de 6 bits, donde cada bit puede ser 0 o 1.

La generación aleatoria de la población inicial es crucial para garantizar la diversidad genética y permitir que el algoritmo explore una amplia variedad de soluciones en el espacio de búsqueda.

Bin	# Valor	# fitness
000010	-0.9365079365079365	0.8770471151423532
111100	0.9047619047619047	0.8185941043083899
000101	-0.8412698412698413	0.7077349458301839
000101	-0.8412698412698413	0.7077349458301839
000111	-0.7777777777777778	0.6049382716049383
110111	0.746031746031746	0.5565633660871756
110110	0.7142857142857142	0.510204081632653
100111	0.23809523809523814	0.056689342403628135
011010	-0.17460317460317465	0.030486268581506694
011110	-0.04761904761904767	0.0022675736961451295

Figura 4: Población Inicial ordenada de manera descendente según su aptitud

### 2.2. Selección

Con nuestros 10 individuos iniciales, se procederá a seleccionar los padres que participarán en la reproducción para generar la siguiente generación. Para esto haremos 5 parejas y cada una generará 2 hijos, para mantener el tamaño de la población constante en 10 individuos.

Separaremos la selección en dos grupos de padres según su aptitud, un grupo será el 40 % de los individuos con mayor aptitud y el otro grupo será el 60 % restante. Para el primer grupo utilizaremos selección por rank (Las que tienen mejor aptitud) y el segundo grupo por torneo (Con peor aptitud esperando que mejoren)

	Bin	# Valor	# fitness
0	000010	-0.9365079365079365	0.8770471151423532
1	111100	0.9047619047619047	0.8185941043083899
2	000101	-0.8412698412698413	0.7077349458301839
3	000101	-0.8412698412698413	0.7077349458301839
	Bin	# Valor	# fitness
4	000111	-0.7777777777777778	0.6049382716049383
5	110111	0.746031746031746	0.5565633660871756
6	110110	0.7142857142857142	0.510204081632653
7	100111	0.23809523809523814	0.056689342403628135
8	011010	-0.17460317460317465	0.030486268581506694
9	011110	-0.04761904761904767	0.0022675736961451295

Figura 5: Grupos de padres por cruza rank (superior) y torneo (inferior)

### 2.2.1. Rank

Para la selección por rank, se ordenan los individuos según su aptitud y se hacen parejas según su rango de aptitud tomando así:  $(1 - 2), (3 - 4), (5 - 6) \dots (n - 1, n)$

Con esto tendremos el siguiente resultado segun nuestro grupo de padres:

```
Pareja 1:
Padre 1 Bin: 000010
Padre 2 Bin: 111100
---
Pareja 2:
Padre 1 Bin: 000101
Padre 2 Bin: 000101
---
Hijo 1 Bin: 000100
Hijo 2 Bin: 111010
Hijo 3 Bin: 000101
Hijo 4 Bin: 000101
```

Figura 6: Padres e hijos generados por cruza rank

Con lo anterior entre padres e hijos, tenemos la siguiente población:

	Bin	# Valor	# fitness
0	000010	-0.9365079365079365	0.8770471151423532
1	111100	0.9047619047619047	0.8185941043083899
2	000100	-0.873015873015873	0.762156714537667
3	000101	-0.8412698412698413	0.7077349458301839
4	000101	-0.8412698412698413	0.7077349458301839
5	000101	-0.8412698412698413	0.7077349458301839
6	000101	-0.8412698412698413	0.7077349458301839
7	111010	0.8412698412698412	0.7077349458301837

Figura 7: Población generada por cruza rank

### 2.2.2. Torneo

Para la selección por torneo, En una lista de  $n$  elementos formaremos parejas la siguiente manera:  $(1 - n), (2 - n - 1), (3 - n - 2) \dots (n/2 - n/2 + 1)$  Con esto tendremos el siguiente resultado segun nuestro grupo de padres:

```
Pareja 1:
Padre 1 Bin: 000111
Padre 2 Bin: 011110
---
Pareja 2:
Padre 1 Bin: 110111
Padre 2 Bin: 011010
---
Pareja 3:
Padre 1 Bin: 110110
Padre 2 Bin: 100111
---
Hijo 1 Bin: 000110
Hijo 2 Bin: 011111
Hijo 3 Bin: 110010
Hijo 4 Bin: 011111
Hijo 5 Bin: 110111
Hijo 6 Bin: 100110
```

Figura 8: Padres e hijos generados por cruza torneo

Con lo anterior entre padres e hijos, tenemos la siguiente población:

	Bin	# Valor	# fitness
0	010010	-0.4285714285714286	0.18367346938775514
1	101011	0.3650793650793651	0.13328294280675235
2	101010	0.33333333333333326	0.11111111111111106
3	101010	0.33333333333333326	0.11111111111111106
4	011010	-0.17460317460317465	0.030486268581506694
5	011100	-0.11111111111111116	0.01234567901234569
6	000110	-0.8095238095238095	0.655328798185941
7	011111	-0.015873015873015928	0.0002519526329050156
8	110010	0.5873015873015872	0.34492315444696386
9	011111	-0.015873015873015928	0.0002519526329050156
10	110111	0.746031746031746	0.5565633660871756
11	100110	0.20634920634920628	0.04257999496094732

Figura 9: Población generada por cruza torneo

### 2.2.3. Selecccion de individuos para la siguiente generación

Con todo lo anterior tenemos una población de 20 individuos (10 por cada tipo de cruza), por lo que debemos seleccionar los 10 mejores individuos para formar la siguiente generación, para esto ordenaremos la población de manera descendente según su aptitud y tomaremos los primeros 10 individuos.



	Bin	# Valor	# fitness
0	000010	-0.9365079365079365	0.8770471151423532
1	111100	0.9047619047619047	0.8185941043083899
2	000100	-0.873015873015873	0.762156714537667
3	000101	-0.8412698412698413	0.7077349458301839
4	000101	-0.8412698412698413	0.7077349458301839
5	000101	-0.8412698412698413	0.7077349458301839
6	000101	-0.8412698412698413	0.7077349458301839
7	111010	0.8412698412698412	0.7077349458301837
8	000110	-0.8095238095238095	0.655328798185941
9	110111	0.746031746031746	0.5565633660871756
10	110010	0.5873015873015872	0.34492315444696386
11	010010	-0.4285714285714286	0.18367346938775514
12	101011	0.3650793650793651	0.13328294280675235
13	101010	0.33333333333333326	0.11111111111111106
14	101010	0.33333333333333326	0.11111111111111106
15	100110	0.20634920634920628	0.04257999496094732
16	011010	-0.17460317460317465	0.030486268581506694
17	011100	-0.11111111111111116	0.01234567901234569
18	011111	-0.015873015873015928	0.0002519526329050156
19	011111	-0.015873015873015928	0.0002519526329050156

Figura 10: Población seleccionada para la siguiente generación

### 2.3. Cruzamiento

Para el cruzamiento se utilizo el método de un punto, donde se selecciona un punto de corte en los cromosomas de los padres y se intercambian las partes posteriores a ese punto. Esto permitirá combinar las características de ambos padres en la descendencia, promoviendo la diversidad genética en la población. En este caso, se seleccionó el punto de corte en la mitad del cromosoma (después del tercer bit) y de esta manera se generaron los hijos de cada pareja de padres.

En este caso no se aplicó mutación, ya que se buscaba observar el comportamiento del algoritmo genético sin la influencia de cambios aleatorios en los individuos. Por lo que una vez hecho la primera generación, se repitió el proceso de selección y cruzamiento para generar nuevas generaciones de manera iterativa. Este proceso se repitió durante 5 generaciones más, observando cómo la aptitud de los individuos mejoraba con el tiempo a medida que se seleccionaban y cruzaban los mejores padres.

Con lo anterior mencionado, destacamos que el criterio de paro fue alcanzar un número máximo de generaciones, en este caso 6 generaciones en total (1 inicial + 5 iteraciones).

### 2.4. Analisis del Algoritmo Genético

Despues de completar las 6 generaciones, se observa una mejora significativa en la aptitud de los individuos a lo largo de las generaciones. La aptitud máxima alcanzada en la última generación es 0.87, lo que indica que el algoritmo ha logrado encontrar soluciones cercanas al óptimo en el espacio de búsqueda definido. Además, la diversidad genética se ha mantenido a lo largo de las generaciones, con una variedad de individuos presentes en la población final.

Esto se puede apreciar en el siguiente gráfico que muestra la evolución de la aptitud máxima y promedio a lo largo de las generaciones:

	Bin	Valor	fitness	Avg_Fitness
0	000010	-0.936508	0.877047	0.487226
1	000010	-0.936508	0.877047	0.720836
2	000010	-0.936508	0.877047	0.785941
3	000010	-0.936508	0.877047	0.842378
4	000010	-0.936508	0.877047	0.877047
5	000010	-0.936508	0.877047	0.877047
6	000010	-0.936508	0.877047	0.877047

Figura 11: Mejor población final de 6 generaciones

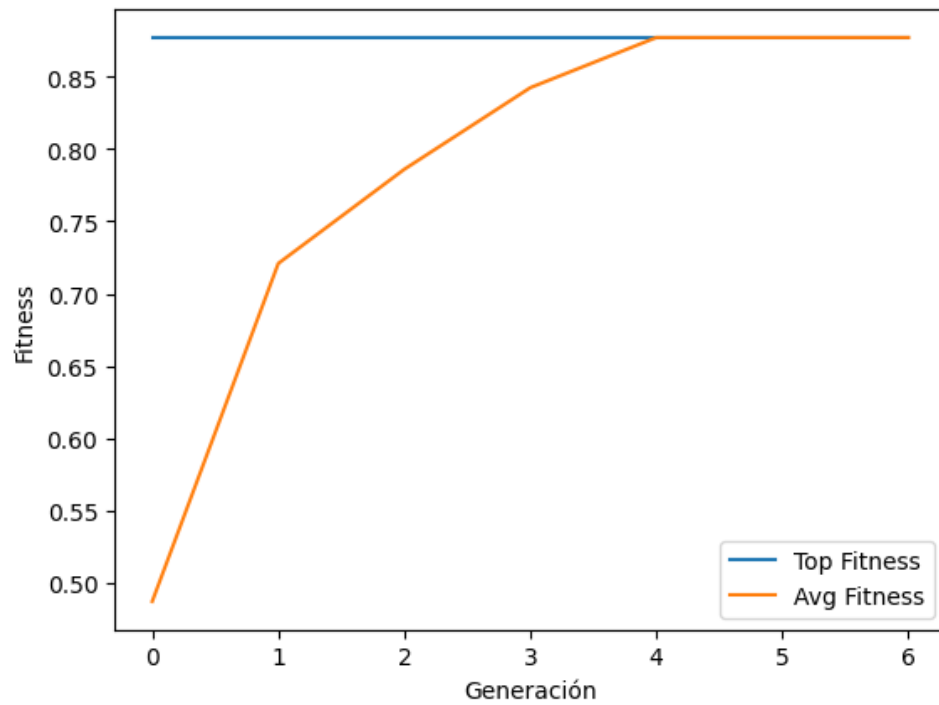


Figura 12: Evolución de la aptitud máxima y promedio a lo largo de las generaciones

aparte de esto graficamos la aptitud en nuestra funcion objetivo  $f(x) = x^2$  para observar como se comporta la aptitud de los individuos en cada generación.

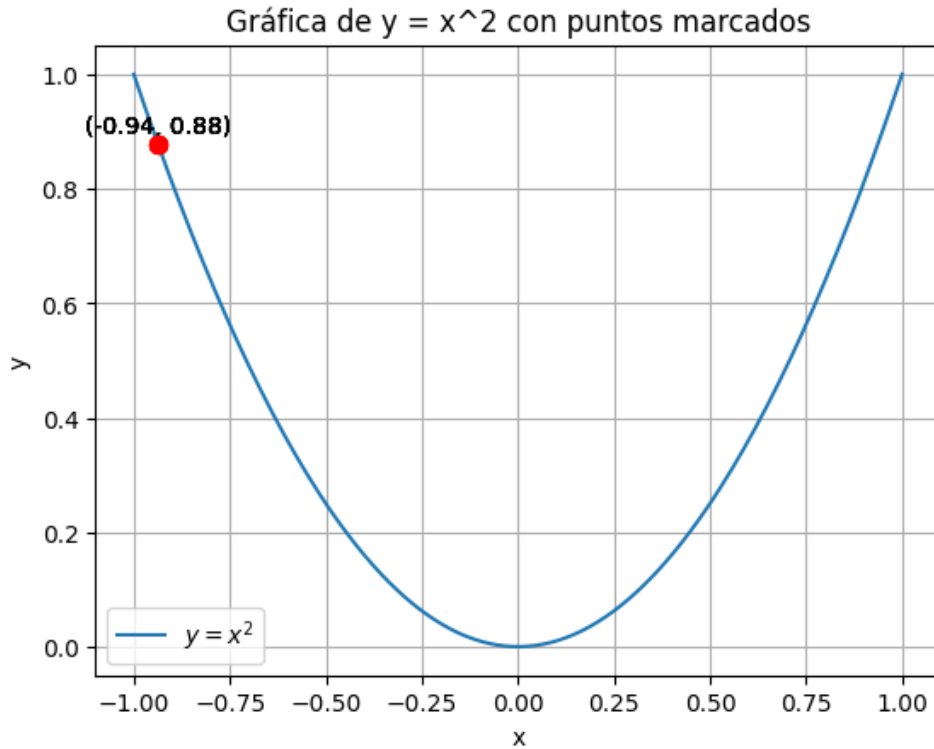


Figura 13: Evolución de la aptitud en la función objetivo  $f(x) = x^2$

En resumen, el algoritmo genético ha demostrado ser efectivo para optimizar la función objetivo  $f(x) = x^2$  dentro del espacio de búsqueda definido.

## 2.5. Segundo Metodo de cruce

Para observar el comportamiento del algoritmo genético con otro método de cruce, se implementó el cruzamiento de dos puntos. En este método, se seleccionan dos puntos de corte en los cromosomas de los padres y se intercambian las partes entre esos puntos. Esto permite una mayor variabilidad en la descendencia y puede ayudar a explorar mejor el espacio de búsqueda.

La única diferencia en la implementación fue el método de cruzamiento, manteniendo el resto del proceso igual, incluyendo la selección por rank y torneo, así como el criterio de paro basado en un número máximo de generaciones (6 en total). Por lo que se repitió el mismo proceso de selección, cruzamiento y selección de individuos para generar nuevas generaciones de manera iterativa.

Entonces nos enfocaremos en mostrar el resultado final de la población después de las 6 generaciones, así como la evolución de la aptitud máxima y promedio a lo largo de las generaciones.

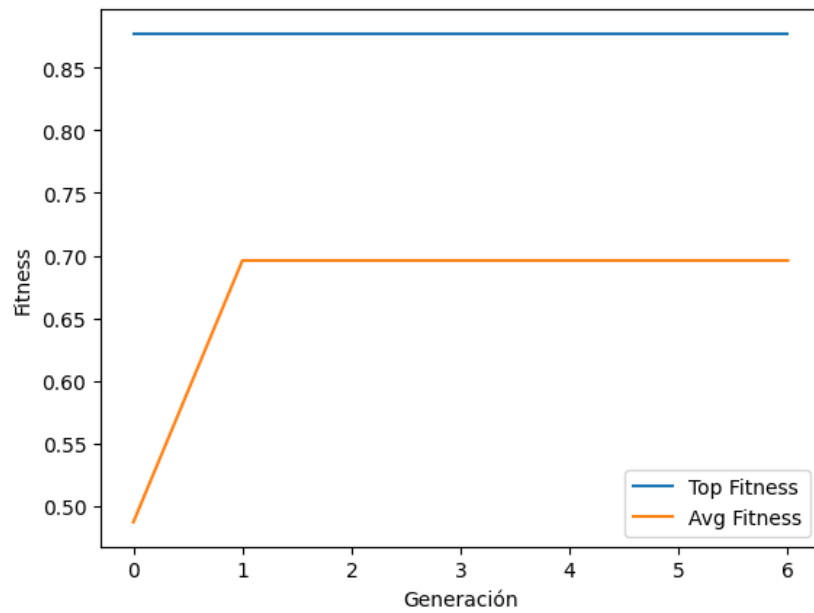


Figura 14: Población final después de 6 generaciones con cruzamiento de dos puntos

### 3. Resultados

En la figura 4 podemos ver la población inicial generada aleatoriamente, donde el individuo con mayor aptitud es 0.87 mientras se desarrolla el AG podemos observar como la aptitud promedio mejora generación tras generación, sin embargo, no siempre el mejor individuo de una generación es mejor que el de la generación anterior, de hecho en este caso como podemos ver más adelante en la figura 11, el mejor individuo de la generacio 0 es decir nuestra población inicial, se mantiene como el mejor individuo durante todas las generaciones.

Esto puede ser un indicativo de que el AG no está explorando lo suficiente el espacio de soluciones, por otro lado esto seria algo esperado ya que no se implementó ninguna técnica de mutacion, por lo que el AG se basa únicamente en la selección y el cruce para generar nuevas soluciones. En la figura 12 podemos observar como la aptitud promedio de la población mejora generación tras generación, pero el maximo se mantiene constante debido a que el mejor individuo de la población inicial se mantiene como el mejor individuo durante todas las generaciones.

Podemos ver que el promedio empieza a converger hacia el maximo, lo que indica que la población está convergiendo hacia una solución óptima, sin embargo, no podemos asegurar que esta solución es la óptima ya que despues de unas generaciones se estanca.

Por otro lado al implementar el segundo metodo de cruce (2 puntos) podemos observar en la figura 14 un comportamiento similar donde el mejor individuo de la población inicial se mantiene como el mejor individuo durante todas las generaciones, sin embargo, es importante destacar que el promedio de la población sube rapidamente en las primeras generaciones y despues se estanca, debido a que la población ya no mejora.

## 4. Discusión

A lo largo del desarrollo del proyecto podemos ver que la implementación correcta de cada etapa es primordial para el correcto funcionamiento del AG esto toma especial relevancia al momento de mencionar que no se implementó la mutación ya que este paso nos permitiría tener un mayor grado de exploración del espacio de soluciones, lo cual podría llevarnos a encontrar mejores soluciones en menos generaciones y de esta manera evitar el estancamiento, además esto podría hacer que nuestro super individuo de la generación 0 se mantenga siempre como el mejor.

Otro punto a destacar es como el cambio en el método de cruce afecta el rendimiento del algoritmo, en este caso el cruce por un punto mostró ser más efectivo que el cruce con dos puntos, esto puede deberse a que al tener un solo punto de corte se mantiene una mayor parte de la información genética de los padres, lo que permite que las soluciones generadas sean más cercanas a las mejores soluciones encontradas hasta el momento. Aunque por otro lado el usar dos puntos hace que la aptitud crezca más rápidamente, pero esto puede llevar a que el algoritmo se estanque en un óptimo local con la misma velocidad como fue el caso.

## 5. Conclusión

Como podemos ver la forma en que el algoritmo busca la solución óptima puede variar mucho dependiendo de los parámetros utilizados, en este caso el método de cruce y la ausencia de mutación afectaron de manera significativa el rendimiento del algoritmo, lo que nos lleva a pensar que la implementación de la mutación podría mejorar los resultados obtenidos, además de que el uso de diferentes métodos de selección y cruce pueden llevar a diferentes resultados, por lo que es importante analizar cada uno de estos parámetros para encontrar la mejor combinación para el problema en cuestión.

Es importante mencionar también el espacio de búsqueda, ya que en este caso es algo pequeño por lo que es útil para entender el funcionamiento del algoritmo, sin embargo, en problemas de mayor complejidad, el espacio de búsqueda puede volverse un factor limitante y es necesario implementar estrategias que permitan una mejor exploración del mismo que abarque una mayor cantidad de soluciones posibles.

## 6. Bibliografía

### Referencias

- [1] Daniel Rivero Gestal et al. *Introducción a los algoritmos genéticos y la programación genética*. Universidade da Coruña, 2010.

.