

# PRÁCTICA 01 - ALGORITMOS GENÉTICO

COMPUTACIÓN NEURONAL Y EVOLUTIVA

PABLO SIMÓN SAINZ  
IVÁN RUIZ GÁZQUEZ

# Índice

---

Representación de los Individuos .....	3
Función de Adaptación .....	4
Experimentos .....	5
COMBINACIÓN BASE.....	5
CAMBIANDO PROBABILIDAD DE CRUCE.....	6
CAMBIANDO PROBABILIDAD DE MUTACIÓN .....	9
CAMBIANDO NÚMERO DE INDIVIDUOS.....	12
MEJOR COMBINACIÓN.....	15
MEJOR COMBINACIÓN CON MÁS GENERACIONES.....	17
Conclusión General .....	20

## Representación de los Individuos

---

Para describir el genoma de cada individuo, utilizaremos una cadena de enteros.

La posición de cada elemento representa el viaje, mientras que el número situado en esta el número del coche que lo efectúa.

Por ejemplo, en un caso en el que hubiera dos coches (0 y 1) y 3 viajes, nos podría quedar un individuo y una población como la siguiente:

```
Individual: [0, 1, 0]
Population:
[0, 1, 1]
[0, 1, 0]
[1, 0, 0]
[1, 0, 1]
[0, 1, 1]
[1, 1, 1]
[1, 0, 0]
[0, 0, 1]
[1, 0, 1]
[0, 0, 1]
```

Ventajas:

- Representación sencilla
- Rápida con el uso del algoritmo

Desventajas:

- No controlas el orden de los viajes dentro del genotipo, por lo que debemos realizar tareas externas para cubrir la falla.

Se busca que con la cantidad de coches dada se efectúen los viajes solicitados, buscando que se cumplan de la mejor manera posible los tiempos.

Nuestro algoritmo facilita una lista con los coches que deberían de efectuar los viajes teniendo en cuenta el cambio de posición.

## Función de Adaptación

En el cálculo de la adaptación, tendremos que ir calculando la puntuación de cada elemento del genoma y sumar el total.

Para calcular esta puntuación individual, realizaremos los siguientes pasos:

1. Comprobamos si el coche llega a tiempo a la posición de inicio, en caso positivo, el coche obtendrá una bonificación.
2. A continuación, haremos lo mismo con el tiempo de llegada.
3. Actualizamos la situación del coche y la puntuación total del genotipo.

La principal variable a tener en cuenta es la posición inicial del coche, que depende del último viaje que haya efectuado.

La posición inicial será lo que decida si el coche llega a tiempo al punto de partida y se lleva la bonificación pertinente.

El no lograr este objetivo, le perjudica a la hora de llegar antes del límite marcado a la meta, ya que parte más tarde de lo previsto.

```
Begin
    Input Genotipo
    paso := 0
    puntuación := 0
    posición := [0, 0]
    For coche, viaje in Genotipo Do
        paso := paso + distancia(posición, origen)

        If paso <= earliest_start Then
            puntuación := puntuación + BONUS
        End If

        paso := paso + distancia(origen, destino)

        If paso <= latest_finish Then
            puntuación := puntuación + distancia(origen, destino)
        End If

        posición := destino

        If paso > PASOS_SIMULACION Then
            break
        End If
    End For
    return puntuación
End
```

## Experimentos

---

### COMBINACIÓN BASE

COMBINACIÓN BASE						
NGEN	CXPB	MUTPB	NIND	TOURNSIZE	INDPB	TIPO CRUCE
50	0.5	0.05	10	3	0.05	OnePoint

Entrada	Fitness
b	171.147
c	9.472.195
e	15.984.482

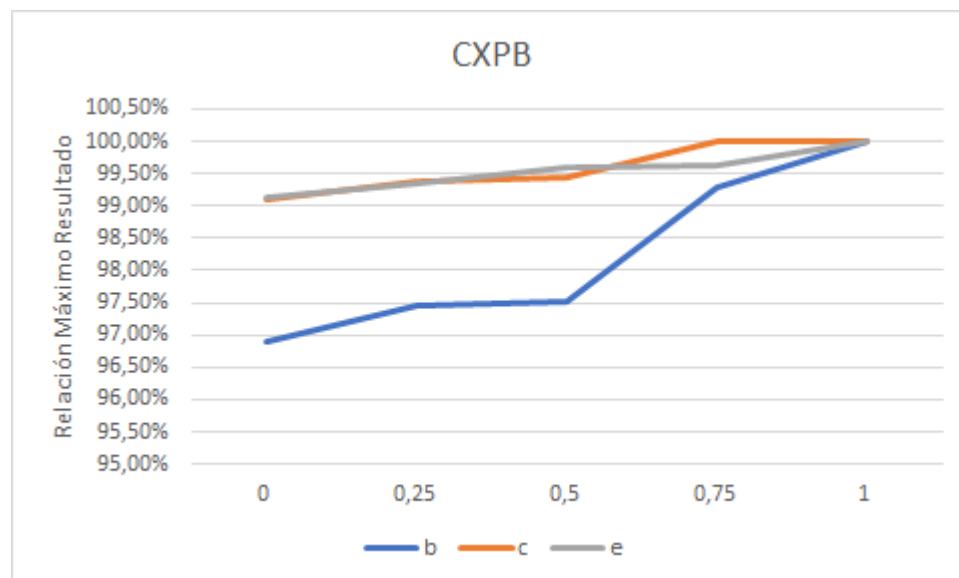
Como combinación base hemos elegido:

- **CXPB medio**, para ir probando de la manera más neutra posible.
- **MUTPB baja**, buscando cierta estabilidad en las gráficas.
- **NGEN y NIND medio-bajo**, para no perder demasiado el tiempo en la espera.

## CAMBIANDO PROBABILIDAD DE CRUCE

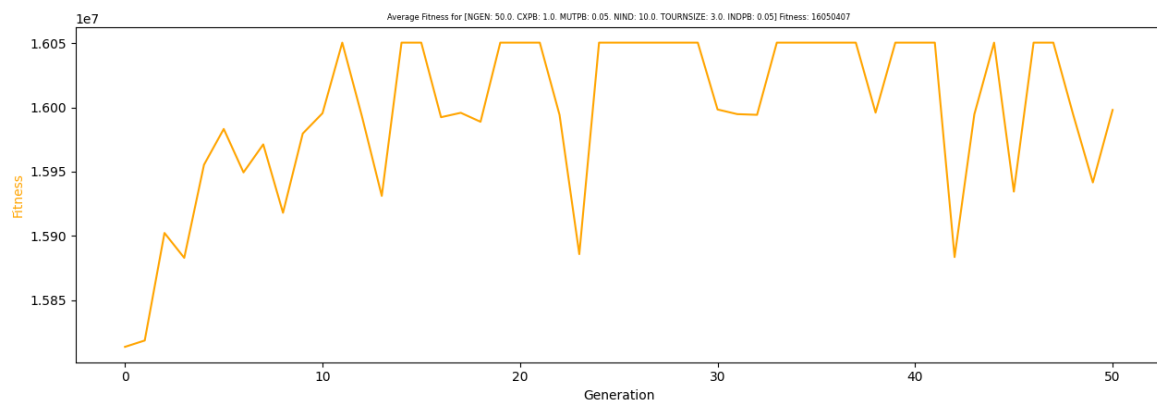
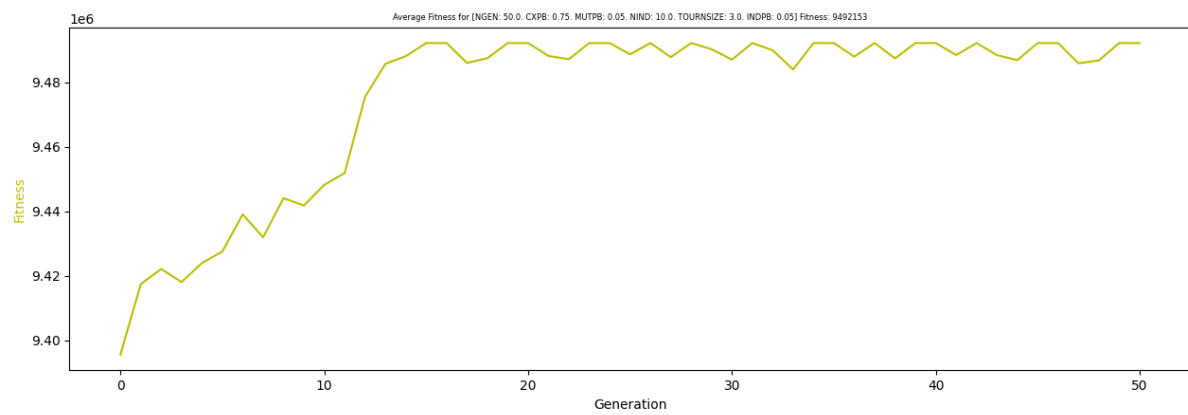
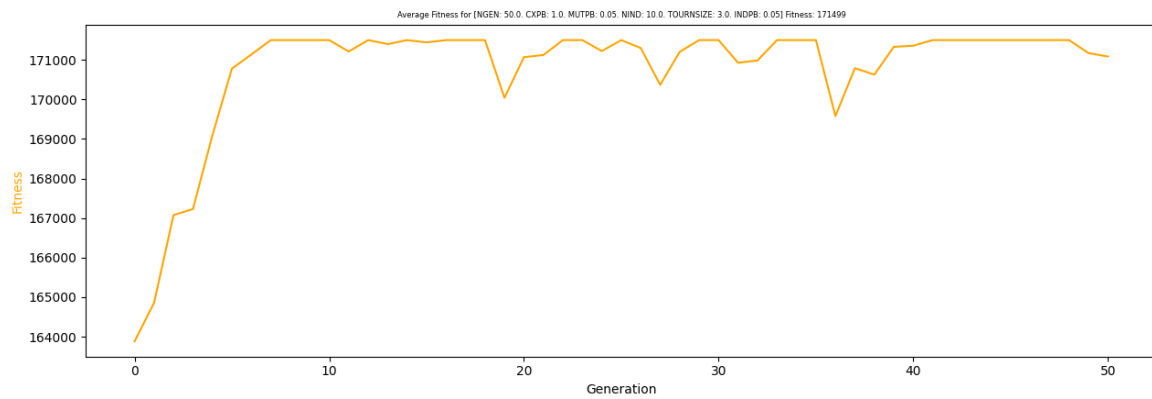
La probabilidad de cruce es la posibilidad que tiene un elemento de que se vea cruzado con otro.

CXPB	0	0,25	0,5	0,75	1
b	168.248	169.231	169.293	172.376	<b>173.626</b>
c	9.408.135	9.432.053	9.439.573	<b>9.492.153</b>	9.491.608
e	15.912.638	15.944.511	15.983.799	15.992.993	<b>16.050.407</b>



Aproximación al máximo resultado obtenido

## Gráficas Cruce



## *Conclusión*

Debido a que el cruce nos permite ir probando combinaciones que desarrollarán hijos con posibles valores mejores que los propios padres, con el objetivo de que estos obtengan los mejores rasgos de sus padres.

Por esto, lo mejor normalmente es tener un valor cercano a 1 (alto) en este parámetro.

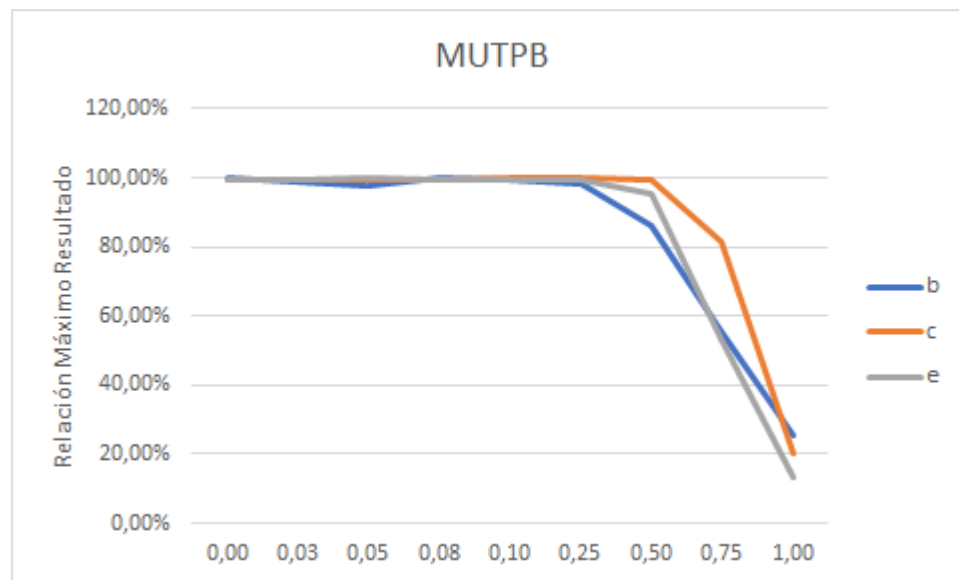


## CAMBIANDO PROBABILIDAD DE MUTACIÓN

Esta variable se encargará de modificar de forma aleatoria una parte del genotipo. Nos es útil para evitar estancarnos en la primera solución que encontremos.

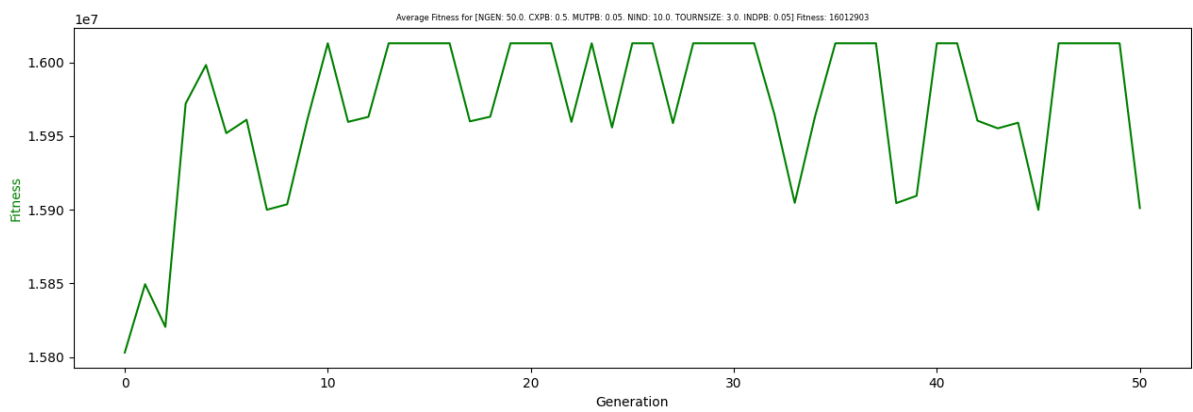
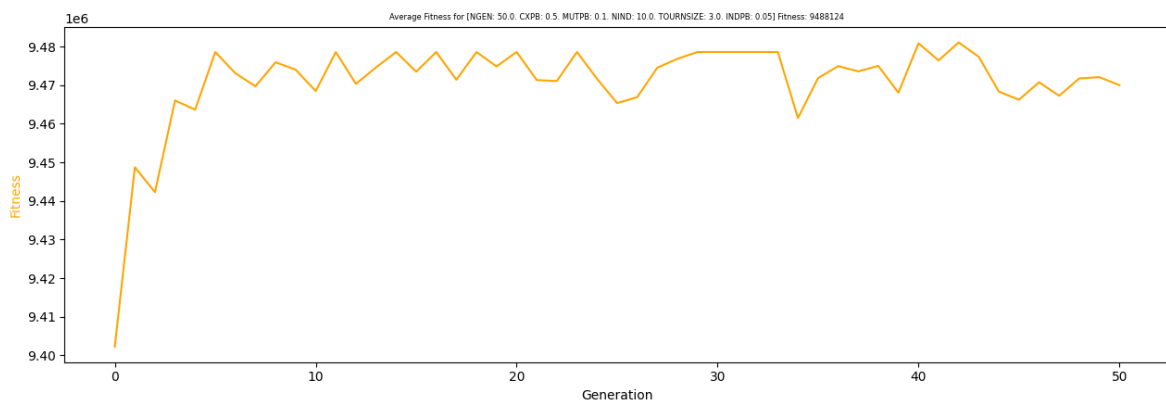
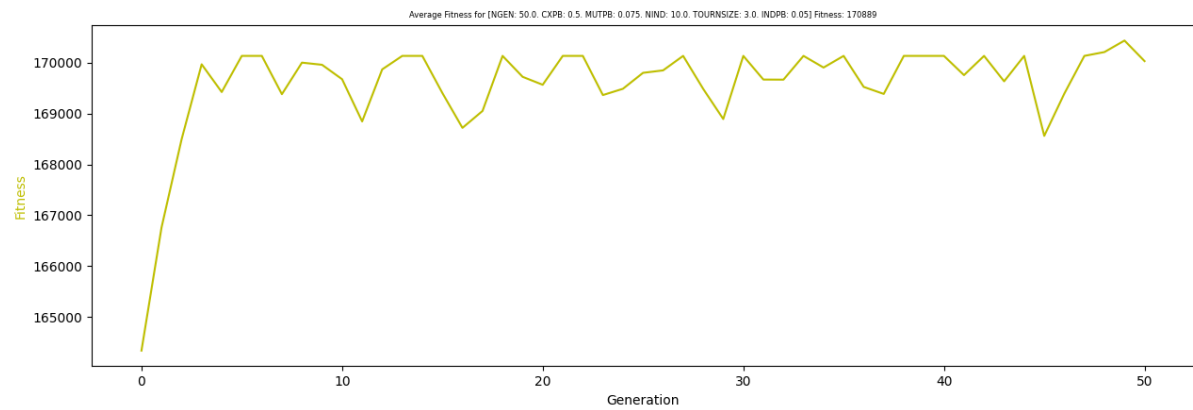
MUTPB	0	0,25	0,5	0,75	1
b	170.696	167.609	147.522	94.775	43.480
c	9.452.556	9.465.313	9.430.076	7.730.230	1.932.724
e	15.937.302	15.950.494	15.276.232	8.533.704	2.105.198

MUTPB	0	0,025	0,05	0,075	0,1
b	170.696	169.257	166.934	<b>170.889</b>	169.600
c	9.452.556	9.449.423	9.453.598	9.456.954	<b>9.488.124</b>
e	15.937.302	15.961.499	<b>16.012.903</b>	15.950.617	15.923.791



Aproximación al máximo resultado obtenido

## Gráficas Mutación



### *Conclusión*

Poner valores altos a esta variable nos dará individuos demasiado aleatorios, por lo que estaremos desaprovechando la selección natural y buscando aleatoriamente resultados.

Sin embargo, si contamos con muchos individuos por generación podremos permitirnos subir ligeramente la mutación.

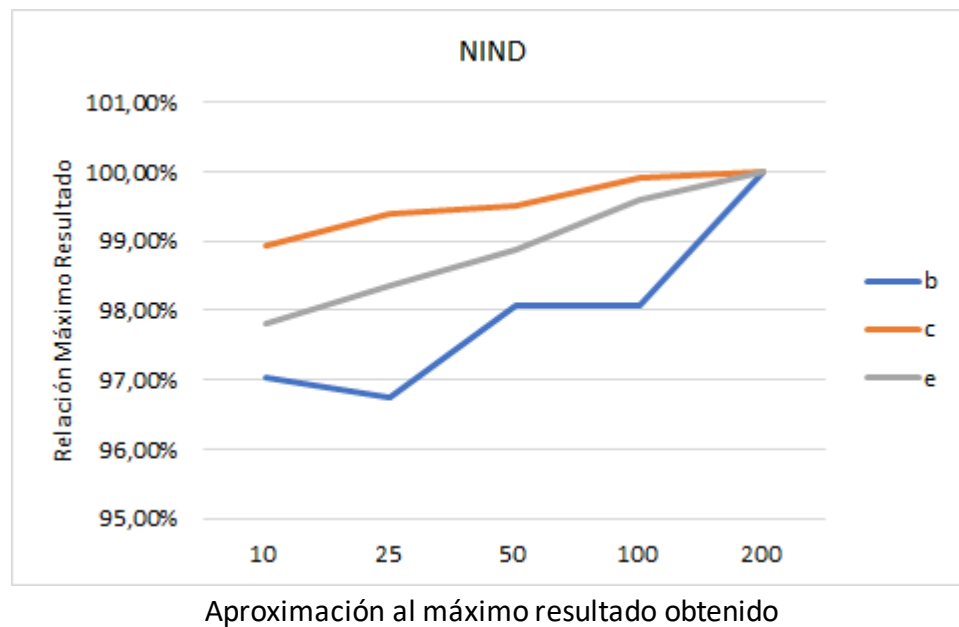
En este caso, una mutación cercana al 7.5% ha sido la más optima.

## CAMBIANDO NÚMERO DE INDIVIDUOS

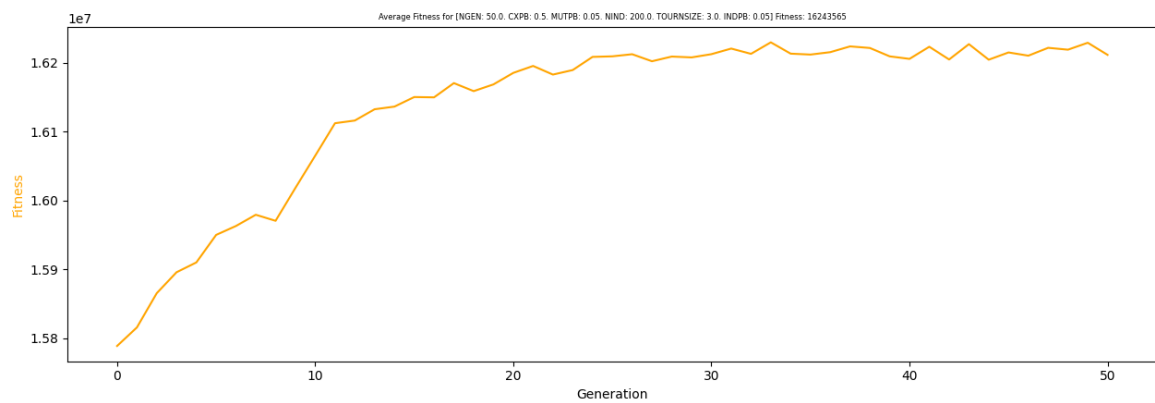
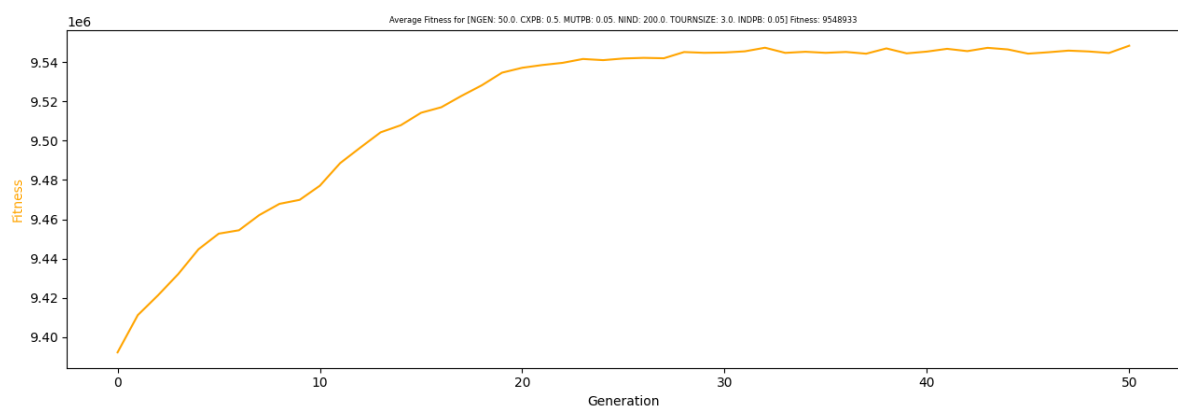
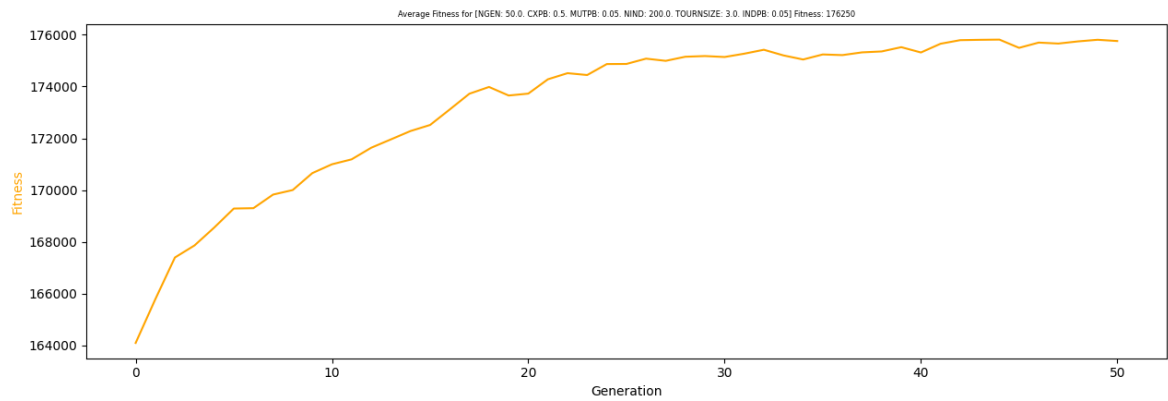
El NGEN es la cantidad de generaciones que obtendremos al usar el algoritmo. Cuanto mayor sea este número, más posibilidades tendremos de encontrar el resultado.

Por otro lado, NIND es la de individuos dentro de una misma generación. A mayores valores, menos generaciones necesitaremos para alcanzar la solución.

NIND	10	25	50	100	200
b	171.029	170.509	172.847	172.847	<b>176.250</b>
c	9.446.005	9.490.059	9.501.201	9.540.067	<b>9.548.933</b>
e	15.887.659	15.975.914	16.062.495	16.176.710	<b>16.243.565</b>



## Gráficas Nº Individuos



## *Conclusión*

Como podemos ver, de primeras parece que queremos tener valores altos para poder obtener la solución lo más temprano y exacto posible.

El problema de tener valores altos en estos parámetros es el tiempo de computación, ya que el número de individuos que estamos analizando es:

$$\text{Individuos totales} = \text{NGEN} * \text{NIND}$$

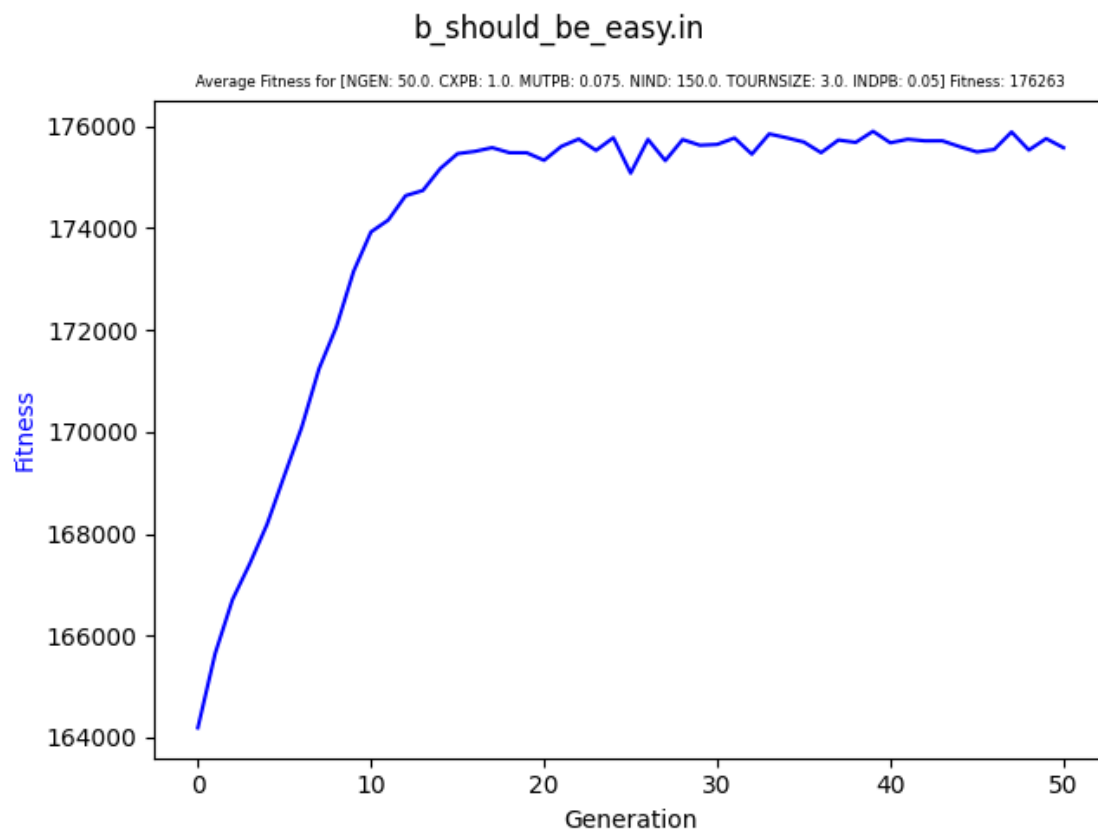
Teniendo en cuenta esto, lo mejor que podemos hacer para dar valores a ambos es tener en cuenta el tiempo de ejecución que nos lleva analizar y modificar cada individuo.

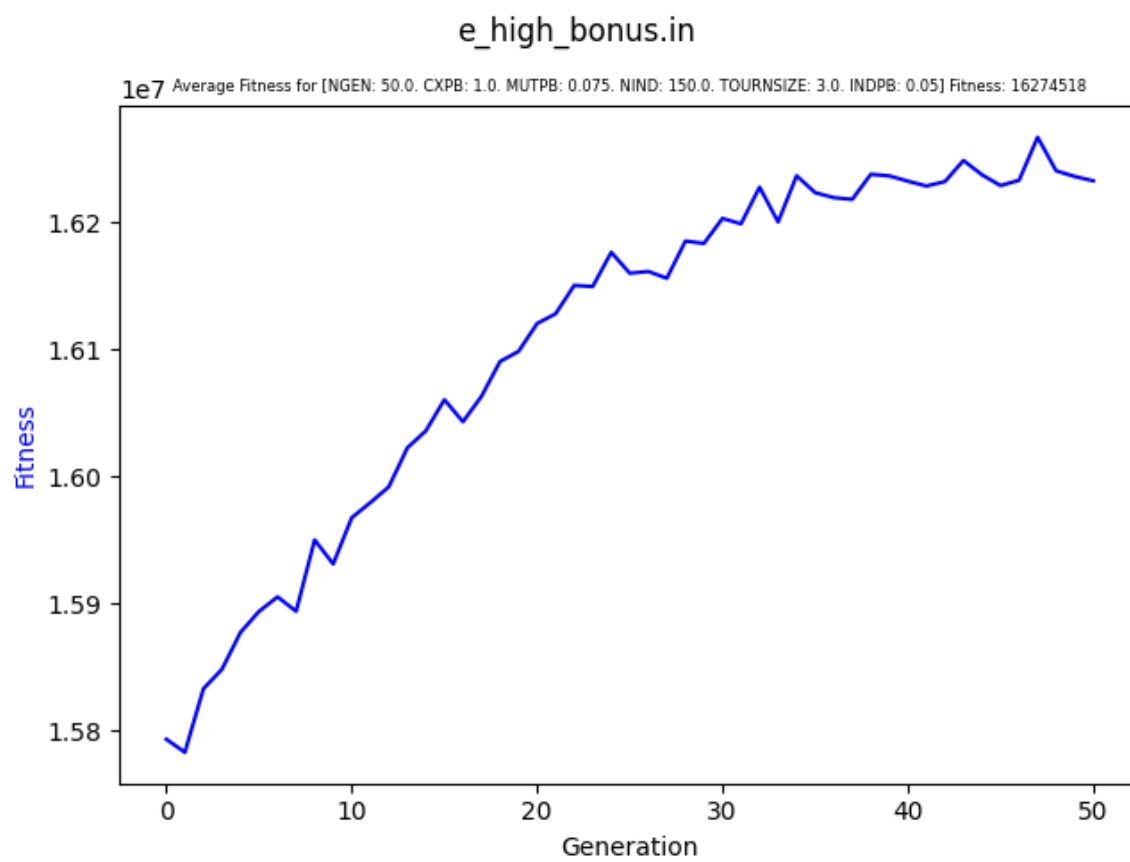
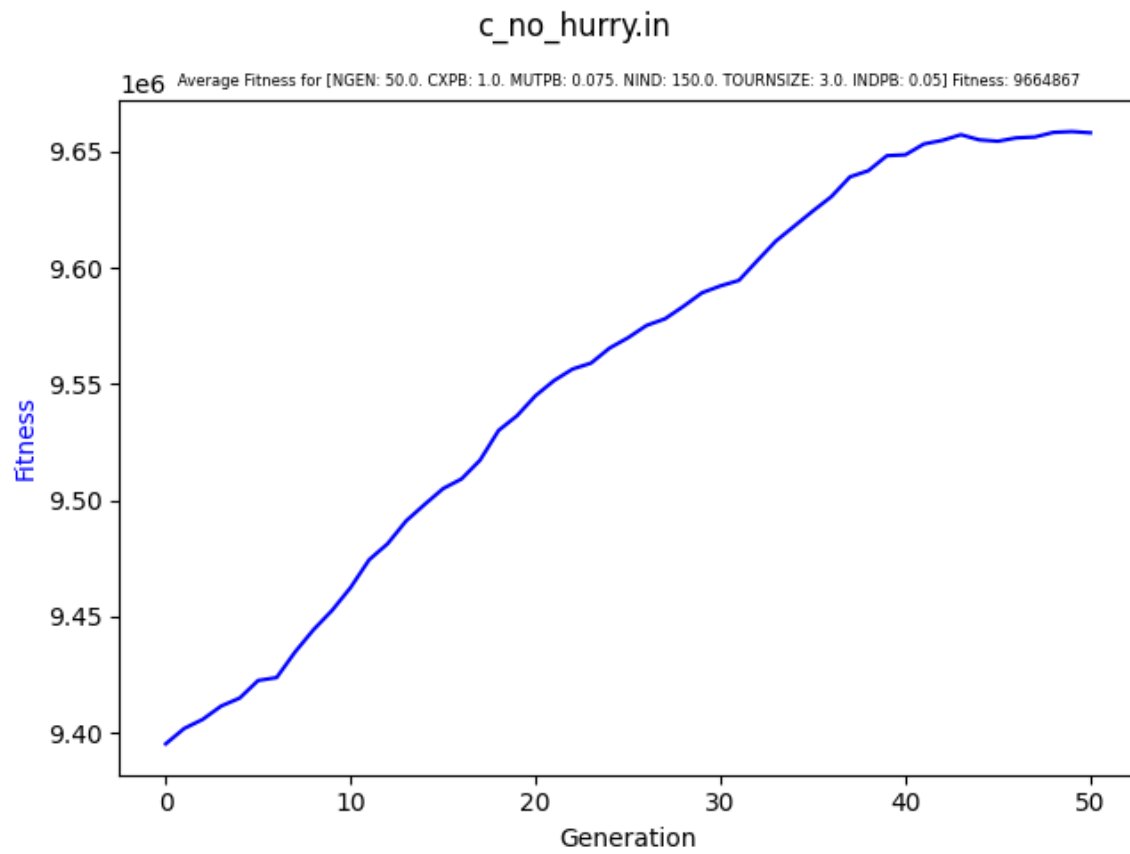
## MEJOR COMBINACIÓN

MEJOR COMBINACIÓN						
NGEN	CXPB	MUTPB	NIND	TOURN SIZE	INDPB	TIPO CRUCE
50	1	0,075	150	3	0.05	OnePoint

Entrada	Fitness
b	176.263
c	9.664.867
e	16.274.518

### Gráficas Mejor Combinación





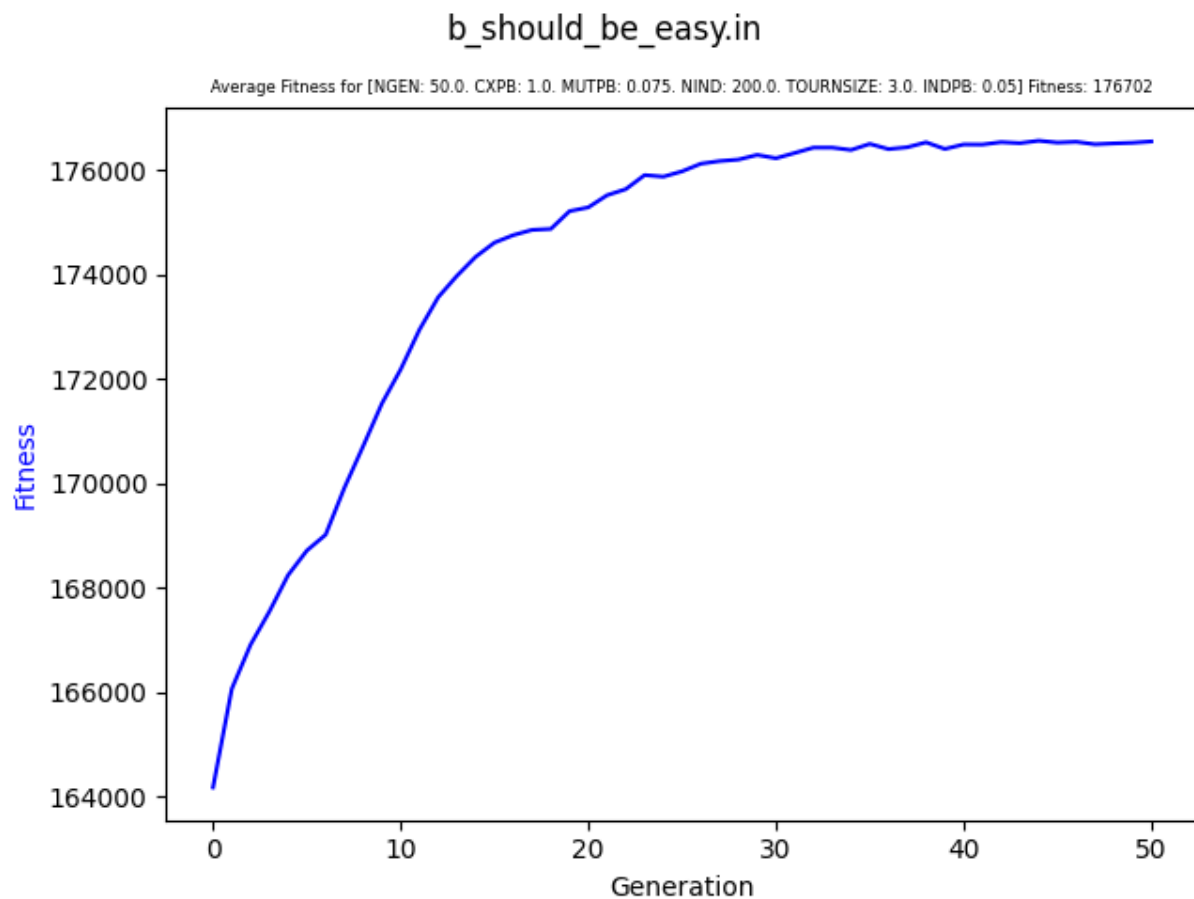


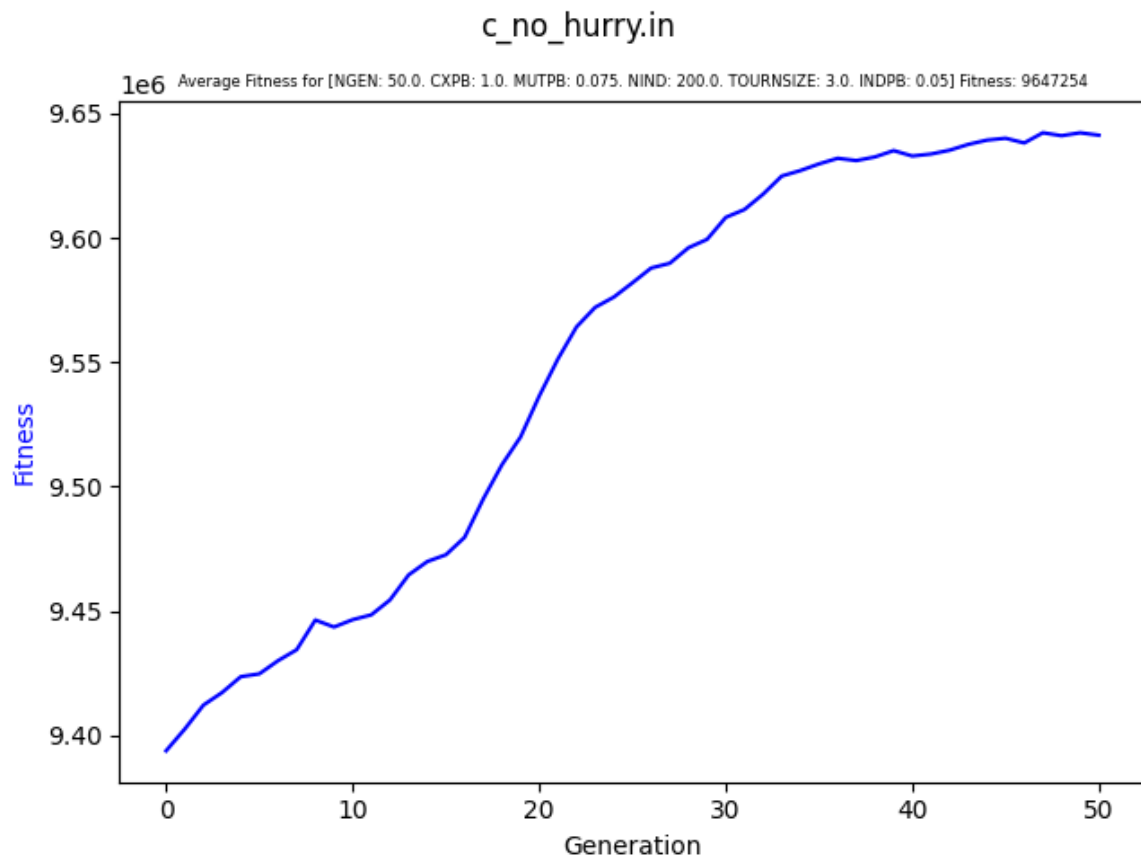
## MEJOR COMBINACIÓN CON MÁS GENERACIONES

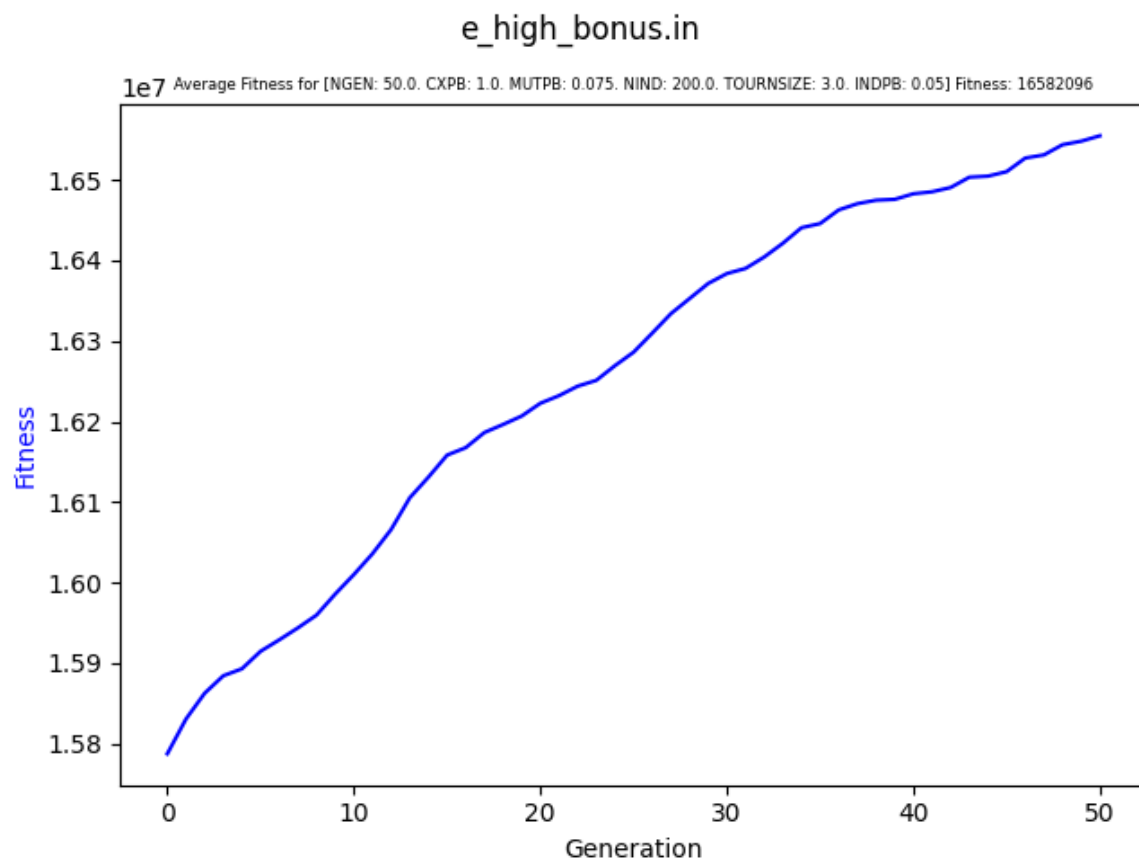
MEJOR COMBINACIÓN						
NGEN	CXPB	MUTPB	NIND	TOURNSIZE	INDPB	TIPO CRUCE
100	1	0,075	150	3	0.05	OnePoint

Entrada	Fitness
b	176.702
c	9.647.245
e	16.582.096

### Gráficas Mejor Combinación







Como podemos observar en esta última gráfica, todavía quedan algunas generaciones más para que el cambio deje de ser significativo.

## Conclusión General

---

Como hemos podido observar, las gráficas con los datos generacionales tienden a tener una forma logarítmica.

Según van avanzando las generaciones nos vamos acercando más hacia nuestra solución, teniendo menor crecimiento a medida que nos acercamos a esta.

El ir jugando con las variables que utilizamos nos ayudará a alcanzar la solución de manera más eficiente, ya que cada problema es diferente y contamos con ciertos recursos computacionales.

Ayudándonos con la gráfica podemos ir cambiando los parámetros poco a poco, si la curva es muy pronunciada, añadimos generaciones, si es demasiado irregular o demasiado poco, ajustamos la mutación, etc.