



## Algoritmos Genéticos

### Trabajo Práctico N°1

Integrantes:

Anmuth, Matias - 50349 - matiasanin15@gmail.com

Koffler, Ignacio - 51739 - nachokoffler@gmail.com

Rigueto, Fabricio - 50553 - fabririgueto@gmail.com

## Indice:

1. Enunciado	2
2. Metodología de desarrollo	3
3. Herramientas de programación	4
4. Metodologías de trabajo	4
5. Codificación	5
6. Graficas	9
7. Parte A	11
8. Parte B	28
9. Conclusion	34

## Enunciado:

Hacer un programa que utilice un Algoritmo Genético Canónico para buscar un máximo de la función:

$$f(x) = (x/(2^{30} - 1))^2 \text{ en el dominio } [0; 2^{30} - 1]$$

Teniendo en cuenta los siguientes datos:

- Probabilidad de Crossover = 0,75
- Probabilidad de Mutación = 0,05
- Población Inicial: 10 individuos
- Ciclos del programa: 20
- Método de Selección: Ruleta
- Método de Crossover: 1 Punto
- Método de Mutación: invertida

Opción A:

- El programa debe mostrar, finalmente, el Cromosoma correspondiente al valor máximo, el valor máximo, mínimo y promedio obtenido de cada población.
- Mostrar la impresión de las tablas de mínimos, promedios y máximos para 20, 100 y 200 corridas.
- Deben presentarse las gráficas de los valores Máximos, Mínimos y Promedios de la función objetivo por cada generación luego de correr el algoritmo genético 20, 100 y 200 iteraciones (una gráfica por cada conjunto de iteraciones)
- Realizar comparaciones de las salidas corriendo el mismo programa en distintos ciclos de corridas y además realizar todos los cambios que considere oportunos en los parámetros de entrada de manera de enriquecer sus conclusiones.

Opción B:

Se entiende por *élite* a un grupo pequeño que por algún motivo, característica, facultad o privilegio es superior o mejor en comparación al grueso de una población determinada; con cualidades o prerrogativas de las que la gran mayoría no disfrutan.

Un algoritmo genético, desde el punto de vista de la optimización, es un método poblacional de búsqueda dirigida basado en probabilidad. *Bajo una condición bastante débil, que el algoritmo mantenga elitismo, es decir, guarde siempre al mejor elemento de la población sin hacerle ningún cambio, se puede demostrar que*

*el algoritmo converge en probabilidad al óptimo. En otras palabras, al aumentar el número de iteraciones, la probabilidad de tener el óptimo en la población tiende a uno.*

Luego el método más utilizado para mejorar la convergencia de los algoritmos genéticos es el elitismo.

Este método consiste básicamente para nuestro trabajo en realizar la etapa de selección de la siguiente manera:

- Se realiza un muestreo en una élite de “ere” miembros es decir para nuestro ejercicio se seleccionan dos cromosomas que poseen el mejor fitness de entre los mejores de la población inicial y se incorporan directamente a la población siguiente, sin pasar por la población intermedia.
- El proceso se repite para cada población que se va generando hasta completar el número de veces que se ejecutará el algoritmo genético. Se solicita la ejecución de 100 iteraciones.

Para esta segunda parte del trabajo se deberá utilizar elitismo, mostrar nuevamente las salidas por pantalla y las gráficas solicitadas en la PARTE A pero en este caso considerando la aplicación de elitismo.

Resolver el ejercicio realizando 100 iteraciones del algoritmo.

## Metodología de desarrollo:

En el programa creamos las siguientes funciones principales para resolver el enunciado:

Creación de la población de cromosomas y asignación de un valor aleatorio a cada genoma de un cromosoma. La cantidad de cromosomas y de genomas por cromosoma están parametrizadas para reutilizar la función en futuros trabajos

Una función que aplica al cromosoma la función objetivo.

Una función fitness para determinar la posibilidad de que un cromosoma sea padre

Dos tipos de ruleta, las dos asignan la probabilidad de que los cromosomas sean padres a su fitness, pero una ordena todos los cromosomas según el fitness para también aplicar elitismo.

Una función de torneo que cada vez que es convocada elige una pareja de padres creando dos torneos en los cuales compiten, en cada uno, dos cromosomas al azar y el mejor gana la posición de padre.

Dos funciones de crossover, ambas tienen una cierta probabilidad de que ocurra el crossover. Una corta en una parte aleatoria a los genomas y mezcla las partes entre sí, creando a los hijos. La otra utiliza una máscara para definir si es el genoma del primer o el del segundo padre el que pasa al hijo.

Tres tipos de mutaciones, una que cambia un índice aleatoriamente, otra que toma una “máscara” con 1s y 0s y cambia los valores de un cromosoma en las posiciones i donde haya un 1 en la máscara, y en la tercera también se crea una máscara pero con valores entre 0 y 1 representando probabilidades las cuales si son mayores que un valor aleatorio “flecha” generado por cada posición.

Dos funciones que setean los usos de los distintos tipos de crossover y mutaciones según si se aplica elitismo o no y que describen las gráficas y las tablas con los valores mínimos, máximos y promedios de cada ciclo.

## Herramientas de programación:

Utilizamos el lenguaje de programación Python ya que los 3 integrantes lo conocemos bastante y nos resultaba cómodo, sumado a eso sabemos que el lenguaje es un estándar tanto en inteligencia artificial como en programación evolutiva (conceptos muy relacionados) gracias a varias funciones integradas y librerías relacionadas. En cuanto a librerías, utilizamos la librería random para varios casos como el de la creación de los genomas relacionados a un cromosoma. Además hicimos uso de la librería matplotlib para graficar cómo cambiaban los cromosomas con valores objetivo máximos, mínimos y promedios entre todos durante una serie de ciclos.

## Metodología de trabajo:

Para trabajar con el código, comúnmente utilizamos la extensión live-share de visual studio code la cual nos permite trabajar en un archivo de manera simultánea y dividir ciertas tareas. El programador host al ejecutar el programa también permite a

los invitados ver la terminal de ejecución y ver tanto el output como los errores que podrían haber surgido.

Sumado a eso utilizamos la plataforma git individualmente con vinculación a un repositorio remoto, de manera que si alguno de nosotros quiere modificar algún aspecto por su cuenta, podrá actualizar su versión sin modificar las anteriores.

## Codificación:

```
#librerias
import random
import matplotlib
import matplotlib.pyplot as plt

#parametros
tam_poblacion = 10
genes = 30
prob_crossover = 0.75
prob_mutacion = 0.05
ciclos = 50 #20 o 100 o 200
tam_torneo = 2
porcentaje_elitismo = 20 #porcentaje
cant_elite = porcentaje_elitismo*tam_poblacion/100

#funciones soporte
def crear_poblacion(numeroDeGenomas, numeroDeIndividuos):
    cromosoma = list()
    cromosomas = list()
    for i in range(numeroDeIndividuos):
        for j in range(numeroDeGenomas):
            cromosoma.append(round(random.random())) #crea un valor aleatorio entre 0 y 1 y lo redonda para el numero entero mas cercano
        cromosomas.append(cromosoma) #lo agrega al array final
        cromosoma = list()
    return cromosomas

def bin_to_dec(cromosoma):
    length = len(cromosoma)
    numDec = 0
    for gen in cromosoma:
        numDec += gen*(2**(length - 1))
        length -= 1
    return numDec #pasaje de binario a decimal. valor que representa el cromosoma pasado como argumento

def fun_obj(cromosoma):
    return (bin_to_dec(cromosoma)/(2**30 - 1))**2 #funcion objetivo

def fitness(cromosoma, tot):
    return fun_obj(cromosoma)/tot #tomamos el total de todos los valores objetivos

def calcular_total(cromosomas):
    tot = 0
    for cromosoma in cromosomas:
        tot += fun_obj(cromosoma)
    return tot #calculamos la suma de los valores objetivo de los cromosomas en cromosomas (el argumento pasado)
```

```

def ordenar(cromosomas):
    tot = calcular_total(cromosomas)
    for i in range(len(cromosomas)):
        for j in range(i, len(cromosomas)):
            if(fitness(cromosomas[i],tot) > fitness(cromosomas[j],tot)):
                aux = cromosomas[j]
                cromosomas[j] = cromosomas[i]
                cromosomas[i] = aux
    return cromosomas #simple ordenamiento segun el fitness de menor a mayor

#selecciones
def ruleta_segun_rango(cromosomas):
    cromosomas = ordenar(cromosomas)#primero ordenamos para poder obtener los indices a partir de los cuales se hace la division de porcentajes
    indicesDeLosPadres=list() #indicesDeLosPadres finalmente sera bidimensional siendo un elemento, un sub array con dos cromosomas padres
    totsum = 0
    for k in range(len(cromosomas)):
        totsum += k + 1 #se suman los indices del array para obtener el total por el cual dividir
    for j in range(int(tam_poblacion/2)):
        indicesDeLosPadres.append([None, None])
        for i in range(2):
            totRuleta = 0
            flecha = random.random()*100 #definimos la flecha, un valor float entre 0 y 100
            index = 0
            condicion = True
            while(condicion):
                totRuleta += ((index + 1)/totsum)*100 #sumamos a totRuleta el porcentaje del indice segun el totsum
                if(totRuleta >= flecha and indicesDeLosPadres[j][0] != index): #si se cumple, significa que flecha cayo en el domino del indice
                    indicesDeLosPadres[j][i] = index
                    condicion = False
                elif(totRuleta >= flecha and indicesDeLosPadres[j][0] == index): #si el padre se repite, se vuelve a tirar la flecha
                    index = 0
                    flecha = random.random()*100
                    totRuleta = 0
                else:
                    index += 1
            return indicesDeLosPadres

```

```

def ruleta_normal(cromosomas):
    total = calcular_total(cromosomas)
    indicesDeLosPadres = list()
    for j in range(int(tam_poblacion/2)):
        indicesDeLosPadres.append([None, None])
        for i in range(2):
            totRuleta = 0
            flecha = random.random()*100 #definimos la flecha, un valor float entre 0 y 100
            index = 0
            condicion = True
            while(condicion):
                totRuleta += (fitness(cromosomas[index], total))*100 #obtenemos el porcentaje que representa el fitness y los sumamos
                if(totRuleta >= flecha and indicesDeLosPadres[j][0] != index):
                    indicesDeLosPadres[j][i] = index
                    condicion = False
                elif(totRuleta >= flecha and indicesDeLosPadres[j][0] == index): #si el padre se repite, se vuelve a tirar la flecha
                    flecha = random.random()*100
                    index=0
                    totRuleta=0
                else:
                    index += 1
            return indicesDeLosPadres

def seleccion_torneo(cromosomas):
    indices_de_padres = list()
    for j in range(int(tam_poblacion/2)):
        for i in range(2): #se hacen dos subgrupos y compiten internamente
            padres = list()
            mejor_indice = 0
            mejor_fitness = 0
            for j in range(tam_torneo): #se evaluan los tam_torneo cromosomas elegidos aleatoriamente
                indice_elegido = random.randint(0,9)
                if (fitness(cromosomas[indice_elegido],calcular_total(cromosomas)) > mejor_fitness):
                    mejor_fitness = fitness(cromosomas[indice_elegido],calcular_total(cromosomas))
                    mejor_indice = indice_elegido
            padres.append(mejor_indice) #se guarda el mejor indice de cada padre en un par
        indices_de_padres.append(padres)
    return indices_de_padres

#funcion para obtener padres
def obtenerpadres(indices, cromosomas, indiceDePareja):
    padres = list()
    for i in range(2):
        padres.append(cromosomas[indices[indiceDePareja][i-1]])
    return padres #devuelve los dos padres del array cromosomas, con indices guardados en indices en la posicion indiceDePareja

```

```

def crossover_mitad_mitad(padres, poblacion):
    if(prob_crossover <= random.random()):
        cruce = random.randint(1, genes - 1)
        hijo1 = padres[0][:cruce] + padres[1][cruce:] #se parte a la mitad cada padre y se dividen las mitades
        hijo2 = padres[1][:cruce] + padres[0][cruce:]
        poblacion.append(hijo1)
        poblacion.append(hijo2)
    else:
        poblacion.append(padres[0]) #se agregan directamente a la nueva poblacion llamada poblacion
        poblacion.append(padres[1])

def crossover_mascara(padres, poblacion_siguiente):
    if(prob_crossover <= random.random()):
        for j in range(2):
            hijo = list()
            mascara = list()
            i = 0
            for i in range(genes):
                mascara.append(round(random.random())) #se define una mascara (cromosoma)
            i = 0
            for i in range(genes):
                if(mascara[i] == 1):
                    hijo.append(padres[0][i]) #si un valor en el indice i en la mascara es 1, el gen i del padre 0 se agrega
                else:
                    hijo.append(padres[1][i]) #si un valor en el indice i en la mascara es 0, el gen i del padre 1 se agrega
            poblacion_siguiente.append(hijo)
    else:
        poblacion_siguiente.append(padres[0])
        poblacion_siguiente.append(padres[1]) #se agregan a la nueva poblacion poblacion_siguiente

#mutaciones
def mutacion_indice_aleatorio(individuo):
    if (prob_mutacion >= random.random()):
        cruce = random.randint(1, genes - 1)
        individuo[cruce] = 1 - individuo[cruce] # Cambiar de 0 a 1 o de 1 a 0 con probabilidad prob_mutacion
    return individuo

def mutacion_mascara(individuo):
    if (prob_mutacion >= random.random()):
        mascara = list()
        i = 0
        for i in range(genes):
            mascara.append(round(random.random()))
        i = 0
        for i in range(genes):
            if(mascara[i] == 1):
                individuo[i] = 1 - individuo[i] #si un valor en el indice i en la mascara es 1, el gen i se modifica
    return individuo

```



```

def crear_universo(metodo_seleccion, metodo_crossover, metodo_mutacion):
    ejex = list()
    valores_minimos = list()
    valores_maximos = list()
    valores_promedio = list()
    poblacion = crear_poblacion(genes, tam_poblacion)
    for j in range(ciclos):
        ejex.append(j)
        valor_promedio = 0
        valor_minimo = 1
        valor_maximo = 0
        tot_sum = 0
        poblacion2 = list() #creamos poblacion siguiente
        padrecitos = metodo_seleccion(poblacion) #obtenemos los indices de los padres
        for k in range(int(tam_poblacion / 2)):
            metodo_crossover(obtenerpadres(padrecitos, poblacion, k), poblacion2) #se hacen, o no, los crossovers
        for i in range(tam_poblacion):
            poblacion2[i] = metodo_mutacion(poblacion2[i]) #mutamos, o no, los cromosomas
            if fun_obj(poblacion2[i]) > valor_maximo:
                valor_maximo = fun_obj(poblacion2[i]) #buscamos el valor maximo en el ciclo
            if fun_obj(poblacion2[i]) < valor_minimo:
                valor_minimo = fun_obj(poblacion2[i]) #buscamos el valor minimo en el ciclo
            tot_sum += fun_obj(poblacion2[i])
        promedio = tot_sum / tam_poblacion #calculamos el valor objetivo promedio en el ciclo
        valores_minimos.append(valor_minimo)
        valores_maximos.append(valor_maximo) #agregamos las estadísticas a un array que las contiene específicamente
        valores_promedio.append(promedio)
        poblacion = poblacion2 #reutilizamos la variable poblacion

```

(aca va lo de las estadísticas)

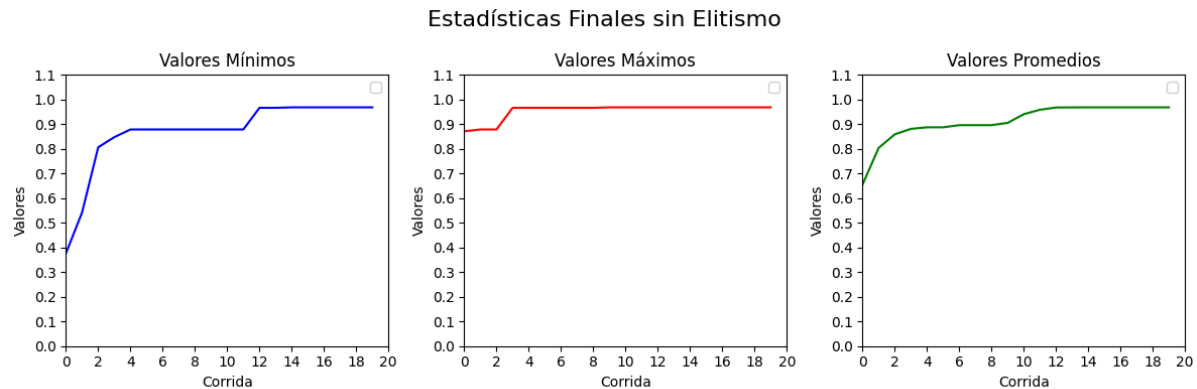
```

#funcion para crear un universo con especificos metodos de desarrollo y que hace uso del elitismo
def crear_universo_con_elitismo(metodo_seleccion, metodo_crossover, metodo_mutacion):
    ejex = list()
    valores_minimos = list()
    valores_maximos = list()
    valores_promedio = list()
    poblacion = crear_poblacion(genes, tam_poblacion)
    for j in range(ciclos):
        ejex.append(j)
        valor_promedio = 0
        valor_minimo = 1
        valor_maximo = 0
        tot_sum = 0
        poblacion2 = list() #creamos poblacion siguiente
        poblacion = ordenar(poblacion)
        for e in range(int(cant_elite)):
            poblacion2.append(poblacion[tam_poblacion - e - 1]) #buscamos los cant_elite mejores cromosomas y los pasamos
        padrecitos = metodo_seleccion(poblacion) #obtenemos los indices de los padres
        for k in range(int((tam_poblacion - cant_elite)/2)):
            metodo_crossover(obtenerpadres(padrecitos, poblacion, k), poblacion2) #se hacen, o no, los crossovers
        for i in range(tam_poblacion):
            poblacion2[i] = metodo_mutacion(poblacion2[i]) #mutamos, o no, los cromosomas
            if fun_obj(poblacion2[i]) > valor_maximo:
                valor_maximo = fun_obj(poblacion2[i]) #buscamos el valor maximo de la poblacion
            if fun_obj(poblacion2[i]) < valor_minimo:
                valor_minimo = fun_obj(poblacion2[i]) #buscamos el valor minimo de la poblacion
            tot_sum += fun_obj(poblacion2[i])
        promedio = tot_sum/tam_poblacion
        valores_minimos.append(valor_minimo)
        valores_maximos.append(valor_maximo) #registramos las estadísticas para un poco mas adelante
        valores_promedio.append(promedio)
        poblacion = poblacion2

```

## Parte A:

### 20 Ciclos:



Metodo de seleccion: ruleta\_segun\_rango, metodo de crossover: crossover\_mitad\_mitad, Metodo de mutacion: mutacion\_indice\_aleatorio

Ciclo	Valor minimo	Valor maximo	Valor promedio	Cromosoma Maximo
-------	--------------	--------------	----------------	------------------

1	0.376462	0.8713851904521712	0.655289	[1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 1, 0, 0, 0, 1, 1, 0, 1, 0, 0, 0, 1, 1, 0]
---	----------	--------------------	----------	--

2	0.542222	0.8786931511748	0.80404	[1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 1, 0, 0, 0, 1, 1, 0, 1, 0, 0, 0, 1, 1, 0]
---	----------	-----------------	---------	--

3	0.80662	0.8786931686349813	0.859037	[1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 1, 0, 0, 0, 1, 1, 0, 1, 0, 1, 0, 0, 0, 0]
---	---------	--------------------	----------	--

4	0.847571	0.9665563049543093	0.881255	[1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0, 1, 1, 1, 0, 1, 1, 0, 0, 1, 1, 0, 1, 0, 1, 1, 0, 0, 0]
---	----------	--------------------	----------	--

5	0.878693	0.9665563049543093	0.887479	[1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0, 1, 1, 1, 0, 1, 1, 0, 0, 1, 1, 0, 1, 0, 1, 1, 0, 0, 0]
---	----------	--------------------	----------	--

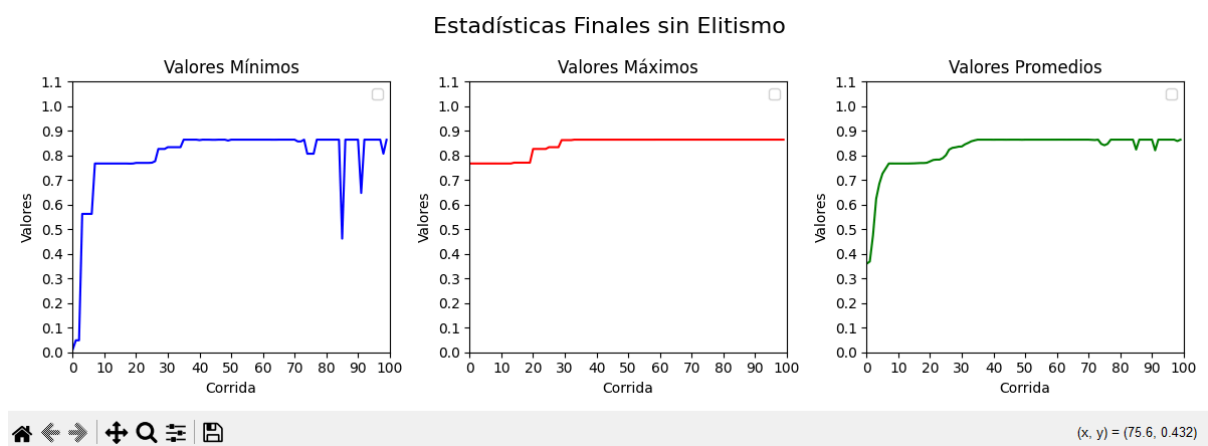
6	0.878693	0.9665563049543093	0.887479	[1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0, 1, 1, 1, 0, 1, 1, 0, 0, 1, 1, 0, 1, 0, 1, 1, 0, 0, 0]
---	----------	--------------------	----------	--

7	0.878693	0.9665563049543093	0.896266
[1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0, 1, 1, 1, 0, 1, 1, 0, 0, 1, 1, 0, 1, 0, 1, 1, 0, 0, 0]			
8	0.878693	0.9665563049543093	0.896266
[1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0, 1, 1, 1, 0, 1, 1, 0, 0, 1, 1, 0, 1, 0, 1, 1, 0, 0, 0]			
9	0.878693	0.9665563049543093	0.896266
[1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0, 1, 1, 1, 0, 1, 1, 0, 0, 1, 1, 0, 1, 0, 1, 1, 0, 0, 0]			
10	0.878693	0.9684774460414728	0.905436
[1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0, 1, 1, 1, 0, 1, 1, 0, 0, 1, 1, 0, 1, 0, 1, 1, 0, 0, 0]			
11	0.878693	0.9684775633568306	0.940774
[1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0, 1, 1, 1, 0, 1, 1, 0, 0, 1, 1, 0, 1, 0, 1, 1, 0, 0, 0]			
12	0.878693	0.9684775633568306	0.958731
[1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0, 1, 1, 1, 0, 1, 1, 0, 0, 1, 1, 0, 1, 0, 1, 1, 0, 0, 0]			
13	0.966556	0.9684775633568306	0.967901
[1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0, 1, 1, 1, 0, 1, 1, 0, 0, 1, 1, 0, 1, 0, 1, 1, 0, 0, 0]			
14	0.966556	0.9684775633568306	0.968285
[1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0, 1, 1, 1, 0, 1, 1, 0, 0, 1, 1, 0, 1, 0, 1, 1, 0, 0, 0]			
15	0.968478	0.9684775780212508	0.968478
[1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0, 1, 1, 1, 0, 1, 1, 0, 0, 1, 1, 0, 1, 0, 1, 1, 0, 0, 0]			
16	0.968478	0.9684775780212508	0.968478
[1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0, 1, 1, 1, 0, 1, 1, 0, 0, 1, 1, 0, 1, 0, 1, 1, 0, 0, 0]			
17	0.968478	0.9684775780212508	0.968478
[1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0, 1, 1, 1, 0, 1, 1, 0, 0, 1, 1, 0, 1, 0, 1, 1, 0, 0, 0]			
18	0.968478	0.9684775780212508	0.968478
[1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0, 1, 1, 1, 0, 1, 1, 0, 0, 1, 1, 0, 1, 0, 1, 1, 0, 0, 0]			
19	0.968478	0.9684775780212508	0.968478
[1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0, 1, 1, 1, 0, 1, 1, 0, 0, 1, 1, 0, 1, 0, 1, 1, 0, 0, 0]			
20	0.968478	0.9684775780212508	0.968478
[1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0, 1, 1, 1, 0, 1, 1, 0, 0, 1, 1, 0, 1, 0, 1, 1, 0, 0, 0]			

El cromosoma con el valor más alto es: [1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0, 1, 1, 1, 0, 1, 1, 0, 0, 1, 1, 0, 1, 0, 1, 1, 0, 0, 0]

100 ciclos:

Metodo de seleccion: ruleta\_segun\_rango, metodo de crossover: crossover\_mitad\_mitad, Metodo de mutacion: mutacion\_indice\_aleatorio



Ciclo	Valor minimo	Valor maximo	Valor promedio	Cromosoma Maximo
-------	--------------	--------------	----------------	------------------

1	0.011464	0.7673280980545811	0.359821	[1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 0, 1, 1, 0, 1, 1, 0, 1, 1, 0, 1, 1, 0, 1, 1, 0, 0, 1, 1, 0, 1, 1]
---	----------	--------------------	----------	---

2	0.048641	0.7673280980545811	0.36903	[1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 0, 1, 1, 0, 1, 1, 0, 1, 1, 0, 1, 1, 0, 0, 1, 1, 0, 1, 1]
---	----------	--------------------	---------	--

3	0.048641	0.7673280980545811	0.47529	[1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 0, 1, 1, 0, 1, 1, 0, 1, 1, 0, 1, 1, 0, 0, 1, 1, 0, 1, 1]
---	----------	--------------------	---------	--

4	0.562476	0.7673280980545811	0.623932	[1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 0, 1, 1, 0, 1, 1, 0, 1, 1, 0, 1, 1, 0, 0, 1, 1, 0, 1, 1]
---	----------	--------------------	----------	--

5	0.562476	0.7673280980545811	0.685387	[1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 0, 1, 1, 0, 1, 1, 0, 1, 1, 0, 1, 1, 0, 0, 1, 1, 0, 1, 1]
---	----------	--------------------	----------	--

6	0.562476	0.7673280980545811	0.726358
	[1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 0, 1, 1, 0, 1, 1, 0, 1, 1, 0, 0, 1, 1, 0, 1, 1]		
7	0.562476	0.7673280915280762	0.746843
	[1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 0, 1, 1, 0, 1, 1, 0, 1, 1, 0, 0, 1, 1, 0, 1, 1]		
8	0.767327	0.7673272561356803	0.767327
	[1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 0, 1, 1, 0, 1, 1, 0, 1, 1, 0, 0, 1, 1, 0, 1, 1]		
9	0.767327	0.7673272561356803	0.767327
	[1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 0, 1, 1, 0, 1, 1, 0, 1, 1, 0, 0, 1, 1, 0, 1, 1]		
10	0.767274	0.7672737919682154	0.767274
	[1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 0, 1, 1, 0, 1, 1, 0, 1, 1, 0, 0, 1, 1, 0, 1, 1]		
11	0.767274	0.7672737919682154	0.767274
	[1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 0, 1, 1, 0, 1, 1, 0, 1, 1, 0, 0, 1, 1, 0, 1, 1]		
12	0.767274	0.7672737919682154	0.767274
	[1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 0, 1, 1, 0, 1, 1, 0, 1, 1, 0, 0, 1, 1, 0, 1, 1]		
13	0.767274	0.7672737919682154	0.767274
	[1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 0, 1, 1, 0, 1, 1, 0, 1, 1, 0, 0, 1, 1, 0, 1, 1]		
14	0.767274	0.7672737919682154	0.767274
	[1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 0, 1, 1, 0, 1, 1, 0, 1, 1, 0, 0, 1, 1, 0, 1, 1]		
15	0.767274	0.7706992537785954	0.767616
	[1, 1, 1, 0, 0, 0, 0, 0, 1, 0, 1, 1, 1, 1, 0, 1, 1, 0, 1, 1, 0, 1, 1, 0, 0, 1, 1, 0, 1, 1]		
16	0.767274	0.7706992537785954	0.767959
	[1, 1, 1, 0, 0, 0, 0, 0, 1, 0, 0, 1, 1, 1, 0, 1, 1, 0, 1, 1, 1, 1, 1, 0, 0, 1, 1, 0, 1, 1]		
17	0.767274	0.7706992537785954	0.768644
	[1, 1, 1, 0, 0, 0, 0, 0, 1, 0, 0, 1, 1, 1, 0, 1, 1, 0, 1, 1, 1, 1, 1, 0, 0, 1, 1, 0, 1, 1]		
18	0.767274	0.7706992537785954	0.769329
	[1, 1, 1, 0, 0, 0, 0, 0, 1, 0, 1, 1, 1, 1, 0, 0, 1, 0, 1, 1, 0, 1, 1, 0, 0, 1, 1, 0, 1, 1]		
19	0.766846	0.7706992537785954	0.769286
	[1, 1, 1, 0, 0, 0, 0, 0, 1, 0, 0, 1, 1, 1, 0, 1, 1, 0, 1, 1, 1, 1, 1, 0, 0, 1, 1, 0, 1, 1]		

20	0.767274	0.7706992537785954	0.770357
[1, 1, 1, 0, 0, 0, 0, 0, 1, 0, 0, 1, 1, 1, 0, 1, 1, 0, 1, 1, 1, 1, 0, 0, 1, 1, 0, 1, 1]			
21	0.769842	0.8265442404028568	0.776024
[1, 1, 1, 0, 1, 0, 0, 0, 1, 0, 1, 1, 1, 1, 0, 1, 1, 0, 1, 1, 1, 1, 0, 0, 1, 1, 0, 1, 1]			
22	0.769842	0.8265442404028568	0.781777
[1, 1, 1, 0, 1, 0, 0, 0, 1, 0, 1, 1, 1, 1, 0, 1, 1, 0, 1, 1, 0, 1, 1, 0, 0, 1, 1, 0, 1, 1]			
23	0.769842	0.8265442404028568	0.783157
[1, 1, 1, 0, 1, 0, 0, 0, 1, 0, 1, 1, 1, 1, 0, 1, 1, 0, 1, 1, 1, 1, 0, 0, 1, 1, 0, 1, 1]			
24	0.769843	0.8265442404028568	0.783157
[1, 1, 1, 0, 1, 0, 0, 0, 1, 0, 1, 1, 1, 1, 0, 1, 1, 0, 1, 1, 1, 1, 0, 0, 1, 1, 0, 1, 1]			
25	0.769843	0.8265442404028568	0.789257
[1, 1, 1, 0, 1, 0, 0, 0, 1, 0, 1, 1, 1, 1, 0, 1, 1, 0, 1, 1, 1, 1, 0, 0, 1, 1, 0, 1, 1]			
26	0.770699	0.8336621928327077	0.801224
[1, 1, 1, 0, 1, 0, 0, 1, 1, 0, 1, 1, 1, 1, 0, 1, 1, 0, 1, 1, 0, 1, 1, 0, 0, 1, 1, 0, 1, 1]			
27	0.776712	0.8336630635862577	0.823697
[1, 1, 1, 0, 1, 0, 0, 1, 1, 0, 1, 1, 1, 1, 0, 1, 1, 0, 1, 1, 1, 1, 0, 0, 1, 1, 0, 1, 1]			
28	0.826545	0.8336630635862577	0.830816
[1, 1, 1, 0, 1, 0, 0, 1, 1, 0, 1, 1, 1, 1, 0, 1, 1, 0, 1, 1, 1, 1, 0, 0, 1, 1, 0, 1, 1]			
29	0.826545	0.8336630635862577	0.832951
[1, 1, 1, 0, 1, 0, 0, 1, 1, 0, 1, 1, 1, 1, 0, 1, 1, 0, 1, 1, 1, 1, 0, 0, 1, 1, 0, 1, 1]			
30	0.826545	0.86243917833393	0.835829
[1, 1, 1, 0, 1, 1, 0, 1, 1, 0, 1, 1, 1, 1, 0, 1, 1, 0, 1, 1, 0, 1, 1, 0, 0, 1, 1, 0, 1, 1]			
31	0.833662	0.8624400639886413	0.836541
[1, 1, 1, 0, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 0, 1, 0, 1, 1, 1, 0, 0, 1, 1, 0, 1, 1]			
32	0.833663	0.8624400639886413	0.845174
[1, 1, 1, 0, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 0, 1, 0, 1, 1, 1, 0, 0, 1, 1, 0, 1, 1]			
33	0.833663	0.8624400639886413	0.850929
[1, 1, 1, 0, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 0, 1, 0, 1, 1, 1, 0, 0, 1, 1, 0, 1, 1]			

34	0.833663	0.8642548389771992	0.857592
[1, 1, 1, 0, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 0, 1, 0, 1, 1, 1, 0, 0, 1, 1, 0, 1, 1]			
35	0.833663	0.8642548389771992	0.861196
[1, 1, 1, 0, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 0, 1, 0, 1, 1, 1, 0, 0, 1, 1, 0, 1, 1]			
36	0.864253	0.8642548389771992	0.864254
[1, 1, 1, 0, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 0, 1, 0, 0, 1, 1, 1, 1, 1, 0, 0, 1, 1, 0, 1, 1]			
37	0.864253	0.8642548389771992	0.864255
[1, 1, 1, 0, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 0, 1, 1, 1, 1, 1, 0, 0, 1, 1, 0, 1, 0]			
38	0.864241	0.8642548389771992	0.864253
[1, 1, 1, 0, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 0, 1, 1, 1, 1, 1, 0, 0, 1, 1, 0, 1, 1]			
39	0.864255	0.8642548389771992	0.864255
[1, 1, 1, 0, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 0, 1, 1, 1, 1, 1, 0, 0, 1, 1, 0, 1, 1]			
40	0.864241	0.8642548389771992	0.864253
[1, 1, 1, 0, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 0, 1, 1, 1, 1, 1, 0, 0, 1, 1, 0, 1, 1]			
41	0.86244	0.8642549498004856	0.864073
[1, 1, 1, 0, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 0, 0, 1, 1, 0, 1, 1, 1, 1, 1, 1, 0, 1, 1, 0, 1, 1]			
42	0.864255	0.8642549498004856	0.864255
[1, 1, 1, 0, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 0, 0, 1, 1, 0, 1, 1, 1, 1, 1, 1, 0, 1, 1, 0, 1, 1]			
43	0.864141	0.8642549498004856	0.864244
[1, 1, 1, 0, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 0, 1, 1, 1, 1, 1, 1, 0, 1, 1, 0, 1, 0]			
44	0.864141	0.8642549498004856	0.864221
[1, 1, 1, 0, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 0, 1, 1, 1, 1, 1, 1, 0, 1, 1, 0, 1, 0]			
45	0.863688	0.8642549498004856	0.864085
[1, 1, 1, 0, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 0, 1, 1, 1, 1, 1, 1, 0, 1, 1, 0, 1, 0]			
46	0.863688	0.8642549498004856	0.864198
[1, 1, 1, 0, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 0, 1, 1, 1, 1, 1, 1, 0, 1, 1, 0, 1, 1]			
47	0.864255	0.8642549480688717	0.864255
[1, 1, 1, 0, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 0, 1, 1, 1, 1, 1, 1, 0, 1, 1, 0, 1, 0]			

48	0.864255	0.8642549480688717	0.864255
[1, 1, 1, 0, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 0, 1, 1, 1, 1, 1, 1, 0, 1, 1, 0, 1, 0]			
49	0.864255	0.8642549480688717	0.864255
[1, 1, 1, 0, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 0, 1, 1, 1, 1, 1, 1, 0, 1, 1, 0, 1, 0]			
50	0.860627	0.8642549480688717	0.863892
[1, 1, 1, 0, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 0, 1, 1, 1, 1, 1, 1, 0, 1, 1, 0, 1, 0]			
51	0.864255	0.8642549480688717	0.864255
[1, 1, 1, 0, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 0, 1, 0, 0, 1, 1, 1, 1, 1, 1, 0, 1, 1, 0, 1, 0]			
52	0.864241	0.8642549480688717	0.864252
[1, 1, 1, 0, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 0, 1, 1, 1, 1, 0, 1, 0, 1, 0, 0, 1, 0]			
53	0.864255	0.8642549480688717	0.864255
[1, 1, 1, 0, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 0, 1, 1, 1, 1, 0, 1, 0, 1, 0, 0, 1, 0]			
54	0.864255	0.8642549480688717	0.864255
[1, 1, 1, 0, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 0, 1, 1, 1, 1, 0, 1, 0, 1, 0, 0, 1, 0]			
55	0.864255	0.8642549480688717	0.864255
[1, 1, 1, 0, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 0, 1, 1, 1, 1, 0, 1, 0, 1, 0, 0, 1, 0]			
56	0.864255	0.8642549480688717	0.864255
[1, 1, 1, 0, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 0, 1, 1, 0, 1, 0]			
57	0.864255	0.8642549480688717	0.864255
[1, 1, 1, 0, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 0, 1, 1, 1, 1, 0, 1, 0, 1, 0, 0, 1, 0]			
58	0.864255	0.8642549480688717	0.864255
[1, 1, 1, 0, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 0, 1, 1, 1, 1, 1, 1, 0, 1, 1, 0, 1, 0]			
59	0.864255	0.8642549480688717	0.864255
[1, 1, 1, 0, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 0, 1, 1, 0, 1, 0]			
60	0.864255	0.8643116905244903	0.864278
[1, 1, 1, 0, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 0, 1, 1, 0, 1, 0]			
61	0.864255	0.8643116905244903	0.864283
[1, 1, 1, 0, 1, 1, 0, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0, 1, 1, 0, 1, 0]			



62	0.864255	0.8643116905244903	0.8643
[1, 1, 1, 0, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0, 1, 1, 0, 1, 0]			
63	0.864255	0.8643116905244903	0.864306
[1, 1, 1, 0, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0, 1, 1, 0, 1, 0]			
64	0.864085	0.8643116905244903	0.864244
[1, 1, 1, 0, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0, 1, 1, 0, 1, 0]			
65	0.864085	0.8643116905244903	0.864289
[1, 1, 1, 0, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0, 1, 1, 0, 1, 0]			
66	0.864312	0.8643116905244903	0.864312
[1, 1, 1, 0, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0, 1, 1, 0, 1, 0]			
67	0.864312	0.8643117459379549	0.864312
[1, 1, 1, 0, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 0]			
68	0.864312	0.8643117459379549	0.864312
[1, 1, 1, 0, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 0]			
69	0.864312	0.8643117459379549	0.864312
[1, 1, 1, 0, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 0]			
70	0.864312	0.8643117459379549	0.864312
[1, 1, 1, 0, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 0]			
71	0.864312	0.8643117459379549	0.864312
[1, 1, 1, 0, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 0]			
72	0.857064	0.8643117459379549	0.863587
[1, 1, 1, 0, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 0]			
73	0.857064	0.8643117459379549	0.862862
[1, 1, 1, 0, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 0]			
74	0.864312	0.8643117459379549	0.864312
[1, 1, 1, 0, 0, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 1, 1, 0, 1, 0]			
75	0.807183	0.8643117459379549	0.847173
[1, 1, 1, 0, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 0]			

76	0.80718	0.8643117459379549	0.841454
[1, 1, 1, 0, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 0]			
77	0.80718	0.8643117459379549	0.84717
[1, 1, 1, 0, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 0]			
78	0.864283	0.8643117459379549	0.864306
[1, 1, 1, 0, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 0]			
79	0.864283	0.8643117459379549	0.864298
[1, 1, 1, 0, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 0]			
80	0.864283	0.8643117459379549	0.864309
[1, 1, 1, 0, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 0]			
81	0.864283	0.8643117459379549	0.864309
[1, 1, 1, 0, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 0]			
82	0.864283	0.8643117459379549	0.864306
[1, 1, 1, 0, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 0]			
83	0.864312	0.8643117459379549	0.864312
[1, 1, 1, 0, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 0]			
84	0.864312	0.8643117459379549	0.864312
[1, 1, 1, 0, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 0]			
85	0.864312	0.8643117459379549	0.864312
[1, 1, 1, 0, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 0]			
86	0.46197	0.8643117459379549	0.824075
[1, 1, 1, 0, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 0]			
87	0.864283	0.8643117459379549	0.864306
[1, 1, 1, 0, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 0]			
88	0.864312	0.8643117459379549	0.864312
[1, 1, 0, 0, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 0]			
89	0.864312	0.8643117459379549	0.864312
[1, 1, 1, 0, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 0]			

90            0.864312            0.8643117459379549            0.864312  
[1, 1, 1, 0, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 0]

91            0.864312            0.8643117459379549            0.864312  
[1, 1, 1, 0, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 0]

92            0.647516            0.8643117459379549            0.820953  
[1, 1, 1, 0, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 0]

93            0.864312            0.8643117459379549            0.864312  
[1, 1, 1, 0, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 0]

94            0.864312            0.8643117459379549            0.864312  
[1, 1, 1, 0, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 0]

95            0.864312            0.8643117459379549            0.864312  
[1, 1, 1, 0, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 0]

96            0.864312            0.8643117459379549            0.864312  
[1, 1, 1, 0, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 0]

97            0.864312            0.8643117459379549            0.864312  
[1, 1, 1, 0, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 0]

98            0.864312            0.8643117459379549            0.864312  
[1, 1, 1, 0, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 0]

99            0.807183            0.8643117459379549            0.858599  
[1, 1, 1, 0, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 0]

100           0.863858            0.8643117459379549            0.864266  
[1, 1, 1, 0, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 0]

El cromosoma con el valor más alto es: [1, 1, 1, 0, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 0, 1, 0]

200 ciclos:

Metodo de seleccion: ruleta\_segun\_rango, metodo de crossover: crossover\_mitad\_mitad, Metodo de mutacion: mutacion\_indice\_aleatorio



Ciclo	Valor minimo	Valor maximo	Valor promedio	Cromosoma Maximo
-------	--------------	--------------	----------------	------------------

1	0.095301	0.7654069950306173	0.500003	[1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 0, 1, 0, 1, 0, 1, 1, 1, 0, 1, 0, 1]
---	----------	--------------------	----------	---

2	0.385648	0.7654069950306173	0.669318	[1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 0, 1, 0, 1, 0, 1, 1, 1, 0, 1, 0, 1]
---	----------	--------------------	----------	---

3	0.520354	0.7654069950306173	0.724098	[1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 0, 1, 0, 1, 0, 1, 1, 1, 0, 1, 0, 1]
---	----------	--------------------	----------	---

4	0.597367	0.7654069950306173	0.748603	[1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 0, 1, 0, 1, 0, 1, 1, 1, 0, 1, 0, 1]
---	----------	--------------------	----------	---

5	0.765407	0.7654069950306173	0.765407	[1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 0, 1, 0, 1, 0, 1, 1, 1, 0, 1, 0, 1]
---	----------	--------------------	----------	---

6	0.765407	0.7654069950306173	0.765407	[1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 0, 1, 0, 1, 0, 1, 1, 1, 0, 1, 0, 1]
---	----------	--------------------	----------	---

7	0.765407	0.7654069950306173	0.765407	[1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 0, 1, 0, 1, 0, 1, 1, 1, 0, 1, 0, 1]
---	----------	--------------------	----------	---

8	0.765407	0.7654069950306173	0.765407	[1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 0, 1, 0, 1, 0, 1, 1, 1, 0, 1, 0, 1]
---	----------	--------------------	----------	---

9	0.765407	0.7654069950306173	0.765407
	[1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0, 1, 0, 1, 0, 1, 1, 1, 0, 1, 0, 1]		
10	0.758587	0.7654069950306173	0.764725
	[1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0, 1, 0, 1, 0, 1, 1, 1, 0, 1, 0, 1]		
11	0.765394	0.9997508494790844	0.78884
	[1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0, 1, 0, 1, 0, 1, 1, 1, 0, 1, 0, 1]		
12	0.765407	0.9997508494790844	0.788841
	[1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0, 1, 0, 1, 0, 1, 1, 1, 0, 1, 0, 1]		
13	0.765407	0.9997508494790844	0.812276
	[1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0, 1, 0, 1, 0, 1, 1, 1, 0, 1, 0, 1]		
14	0.765407	0.9997508494790844	0.83571
	[1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0, 1, 0, 1, 0, 1, 1, 1, 0, 1, 0, 1]		
15	0.765407	0.9997508494790844	0.906013
	[1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0, 1, 0, 1, 0, 1, 1, 1, 0, 1, 0, 1]		
16	0.765407	0.9997508494790844	0.952876
	[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0, 1, 0, 1, 0, 1, 1, 1, 0, 1, 0, 1]		
17	0.999721	0.9997518030348175	0.999746
	[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0, 1, 1, 1, 0, 1, 1, 1, 0, 1, 0, 1]		
18	0.995834	0.9997518030348175	0.999359
	[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0, 1, 1, 1, 0, 1, 1, 1, 0, 1, 0, 1]		
19	0.938235	0.9997518030348175	0.987448
	[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0, 1, 1, 1, 0, 1, 1, 1, 0, 1, 0, 1]		
20	0.999751	0.9997518030348175	0.999751
	[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0, 1, 1, 1, 0, 1, 1, 1, 0, 1, 0, 1]		
21	0.999751	0.9997518030348175	0.999751
	[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0, 1, 1, 1, 0, 1, 1, 1, 0, 1, 0, 1]		
22	0.999751	0.9997518030348175	0.999751
	[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0, 1, 1, 1, 0, 1, 1, 1, 0, 1, 0, 1]		

23	0.999751	0.9997518030348175	0.999751
[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0, 1, 1, 1, 0, 1, 1, 1, 0, 1, 0, 1]			
24	0.999751	0.9997518030348175	0.999751
[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0, 1, 1, 1, 0, 1, 1, 1, 0, 1, 0, 1]			
25	0.999751	0.9997518030348175	0.999751
[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0, 1, 1, 1, 0, 1, 1, 1, 0, 1, 0, 1]			
26	0.997799	0.9997518030348175	0.999556
[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0, 1, 1, 1, 0, 1, 1, 1, 0, 1, 0, 1]			
27	0.999751	0.9997518030348175	0.999752
[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0, 1, 1, 1, 0, 1, 1, 1, 0, 1, 0, 1]			
28	0.99585	0.9997518030348175	0.998581
[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0, 1, 1, 1, 0, 1, 1, 1, 0, 1, 0, 1]			
29	0.99585	0.9997518030348175	0.999362
[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0, 1, 1, 1, 0, 1, 1, 1, 0, 1, 0, 1]			
30	0.99585	0.9997518030348175	0.999362
[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0, 1, 1, 1, 0, 1, 1, 1, 0, 1, 0, 1]			
31	0.96875	0.9997518030348175	0.993551
[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 0, 1, 1, 1, 0, 1, 1, 1, 0, 1, 1, 1, 0, 1, 0, 1]			
32	0.96875	0.9997518030348175	0.996652
[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0, 1, 1, 1, 0, 1, 1, 1, 0, 1, 0, 1]			
33	0.999752	0.9997518030348175	0.999752
[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 0, 1, 1, 1, 1, 1, 0, 1, 1, 1, 0, 1, 1, 1, 0, 1, 0, 1]			
34	0.999691	0.9997518030348175	0.99974
[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0, 1, 1, 1, 0, 1, 1, 1, 0, 1, 0, 1]			
35	0.999691	0.9997518030348175	0.999746
[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0, 1, 1, 1, 0, 1, 1, 1, 0, 1, 0, 1]			
36	0.999752	0.9997518030348175	0.999752
[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 0, 1, 1, 1, 1, 1, 0, 1, 1, 1, 0, 1, 1, 1, 0, 1, 0, 1]			

37	0.99585	0.9997518030348175	0.998971
[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0, 1, 1, 1, 0, 1, 1, 0, 1, 0, 1]			
38	0.99585	0.9997518030348175	0.999362
[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 0, 1, 0, 1]			
39	0.998776	0.9997518030348175	0.999459
[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0, 1, 1, 1, 0, 1, 1, 0, 1, 0, 1]			
40	0.998776	0.9997518030348175	0.999459
[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0, 1, 1, 1, 0, 1, 1, 0, 1, 0, 1]			
41	0.999752	0.9997518030348175	0.999752
[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0, 1, 1, 1, 0, 1, 1, 0, 1, 0, 1]			
42	0.998776	0.9997556172622971	0.999459
[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 0, 1, 0, 1]			
43	0.998776	0.9997556172622971	0.999558
[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 0, 1, 0, 1]			
44	0.999751	0.9997556321616374	0.999754
[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1]			
45	0.999752	0.9997556321616374	0.999754
[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1]			
46	0.999756	0.9997556321616374	0.999756
[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 0, 1]			
47	0.999756	0.9997556321616374	0.999756
[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 0, 1]			
48	0.999748	0.9997556321616374	0.999755
[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 0, 1]			
49	0.998779	0.9997558705510983	0.999658
[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1]			
50	0.968754	0.9997558705510983	0.996558
[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1]			

51	0.999756	0.9997558705510983	0.999756
[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1]			
52	0.999756	0.9997558705510983	0.999756
[1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 0, 1]			
53	0.999268	0.9997558705510983	0.999705
[1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1]			
54	0.999756	0.9997558705510983	0.999756
[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1]			
55	0.999756	0.9997558705510983	0.999756
[1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1]			
56	0.999756	0.9997558705510983	0.999756
[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1]			
57	0.968754	0.9997558705510983	0.971854
[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1]			
58	0.968746	0.9997558705510983	0.97495
[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1]			
59	0.968746	0.9997558705510983	0.981153
[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1]			
60	0.968746	0.9997558705510983	0.981154
[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1]			
61	0.968754	0.9997558705510983	0.984255
[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1]			
62	0.968754	0.9997558705510983	0.993507
[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 0, 1]			
63	0.738316	0.9997558705510983	0.973612
[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 1]			
64	0.738316	0.9997558705510983	0.973612
[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1]			



65	0.738316	0.9997558705510983	0.973612
[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1]			
66	0.999756	0.9997558705510983	0.999756
[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1]			
67	0.999756	0.9997558705510983	0.999756
[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1]			
68	0.999756	0.9997558705510983	0.999756
[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1]			
69	0.999756	0.9997558705510983	0.999756
[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1]			
70	0.999756	0.9997558705510983	0.999756
[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1]			
71	0.99878	0.9997558705510983	0.999463
[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1]			
72	0.99878	0.9997558705510983	0.999658
[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1]			
73	0.99878	0.9997558705510983	0.999658
[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1]			
74	0.999756	0.9997558705510983	0.999756
[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1]			
75	0.999756	0.9997558705510983	0.999756
[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1]			
76	0.999748	0.9997558705510983	0.999755
[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1]			
77	0.999725	0.9997558705510983	0.999753
[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 0, 1]			
78	0.999237	0.9997558705510983	0.9996
[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1]			

79	0.999237	0.9997558705510983	0.999652
[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1]			
80	0.999237	0.9997558705510983	0.999704
[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1]			
81	0.999756	0.9997558705510983	0.999756
[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1]			
82	0.999756	0.9997558705510983	0.999756
[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1]			
83	0.765411	0.9997558705510983	0.976318
[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1]			
84	0.999756	0.9997558705510983	0.999756
[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1]			
85	0.999756	0.9997558705510983	0.999756
[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1]			
86	0.999634	0.9997558705510983	0.999658
[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1]			
87	0.991838	0.9997558705510983	0.998891
[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1]			
88	0.999634	0.9997558705510983	0.999683
[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1]			
89	0.999634	0.9997558705510983	0.999658
[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1]			
90	0.999634	0.9997558705510983	0.999658
[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1]			
91	0.999634	0.9997558705510983	0.999646
[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1]			
92	0.999626	0.9997558705510983	0.999656
[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1]			

93	0.999626	0.9997558705510983	0.99967
[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1]			
94	0.999634	0.9997558705510983	0.999719
[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1]			
95	0.995854	0.9997558705510983	0.998963
[1, 1, 1, 1, 0, 1, 1, 1, 0, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0]			
96	0.995854	0.9997558705510983	0.998963
[1, 1, 1, 1, 0, 1, 1, 1, 0, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0]			
97	0.999634	0.9997558686886805	0.999744
[1, 1, 1, 1, 0, 1, 1, 1, 0, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0]			
98	0.93446	0.9997558686886805	0.947519
[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0]			
99	0.93446	0.9997558686886805	0.954049
[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0]			
100	0.934456	0.9997558686886805	0.973637
[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0]			
101	0.51362	0.9997558686886805	0.951142
[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0]			
102	0.999756	0.9997558686886805	0.999756
[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0]			
103	0.999756	0.9997558686886805	0.999756
[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0]			
104	0.999756	0.9997558686886805	0.999756
[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0]			
105	0.999756	0.9997558686886805	0.999756
[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0]			
106	0.999756	0.9997558686886805	0.999756
[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0]			

107	0.999756	0.9997558686886805	0.999756
[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0]			
108	0.99196	0.9997558686886805	0.998967
[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0]			
109	0.999741	0.9997558686886805	0.999754
[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0]			
110	0.999756	0.9997558686886805	0.999756
[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0]			
111	0.999756	0.9997558686886805	0.999756
[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0]			
112	0.999634	0.9997558686886805	0.999744
[1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 0, 0]			
113	0.93824	0.9997558686886805	0.9813
[1, 0, 0]			
114	0.93824	0.9997558686886805	0.993604
[1, 0, 0]			
115	0.999756	0.9997558686886805	0.999756
[1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0, 0]			
116	0.968754	0.9999999944120646	0.990504
[1, 0, 0]			
117	0.968754	0.9999999944120646	0.99668
[1, 0, 0]			
118	0.984194	0.9999999944120646	0.996717
[1, 0, 0]			
119	0.984194	0.9999999944120646	0.998297
[1, 0, 0]			
120	0.999756	0.9999999944120646	0.999902
[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0]			

[illegible]

135	1.0	0.9999999944120646	1.0
[1, 0, 0]			
136	1.0	0.9999999944120646	1.0
[1, 0, 0]			
137	1.0	0.9999999944120646	1.0
[1, 0, 0]			
138	1.0	0.9999999944120646	1.0
[1, 0, 0]			
139	1.0	0.9999999944120646	1.0
[1, 0, 0]			
140	1.0	0.9999999944120646	1.0
[1, 0, 0]			
141	1.0	0.9999999944120646	1.0
[1, 0, 0]			
142	0.999939	0.9999999944120646	0.999994
[1, 0, 0]			
143	0.996098	0.9999999944120646	0.999213
[1, 0, 0]			
144	0.999939	0.9999999944120646	0.999994
[1, 0, 0]			
145	1.0	0.9999999962747097	1.0
[1, 0, 1]			
146	1.0	0.9999999962747097	1.0
[1, 0, 1]			
147	1.0	0.9999999962747097	1.0
[1, 0, 1]			
148	1.0	0.9999999962747097	1.0
[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1]			

149        1.0            0.9999999962747097        1.0  
[1, 0, 1]

150        1.0            0.9999999962747097        1.0  
[1, 0, 1]

151        0.999969        0.9999999962747097        0.999997  
[1, 0, 1]

152        1.0            0.9999999962747097        1.0  
[1, 1, 1, 0, 1, 0, 1]

153        1.0            0.9999999962747097        1.0  
[1, 1, 1, 0, 1, 0, 1]

154        0.878906        0.9999999962747097        0.963672  
[1, 0, 1]

155        0.878906        0.9999999962747097        0.975781  
[1, 0, 1]

156        1.0            0.9999999962747097        1.0  
[1, 0, 1]

157        1.0            0.9999999962747097        1.0  
[1, 0, 1]

158        1.0            0.9999999962747097        1.0  
[1, 0, 1]

159        1.0            0.9999999962747097        1.0  
[1, 0, 1]

160        1.0            0.9999999962747097        1.0  
[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1]

161        0.992203        0.9999999962747097        0.99922  
[1, 0, 1]

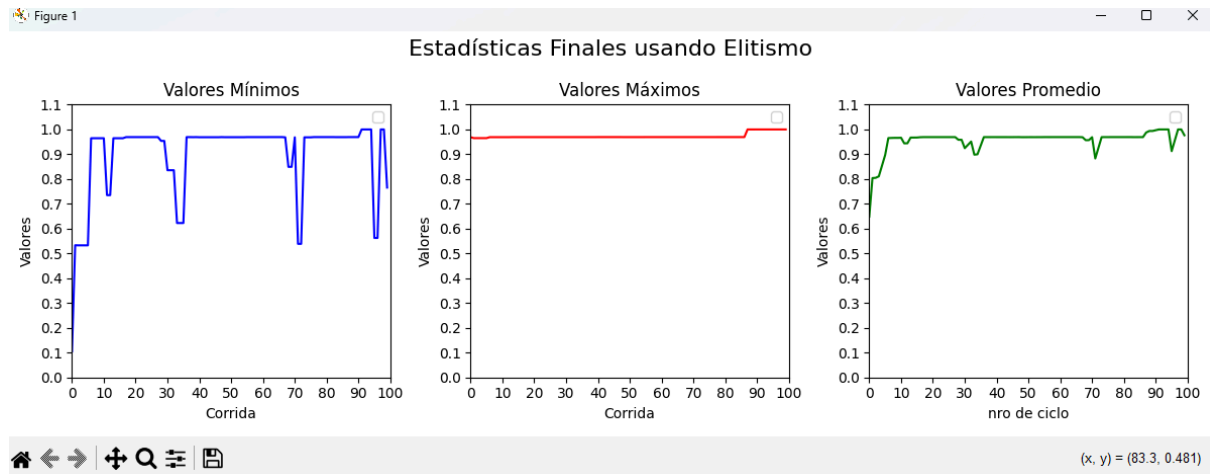
162        0.999939        0.9999999962747097        0.999988  
[1, 0, 1]

163	0.999985	0.9999999962747097	0.999995
[1, 0, 1]			
164	0.999985	0.9999999962747097	0.999998
[1, 0, 1]			
165	0.999985	0.9999999962747097	0.999998
[1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1]			
166	0.984436	0.9999999962747097	0.998444
[1, 0, 1]			
167	0.984436	0.9999999962747097	0.998444
[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1]			
168	1.0	0.9999999962747097	1.0
[1, 0, 1]			
169	1.0	0.9999999962747097	1.0
[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1]			
170	1.0	0.9999999962747097	1.0
[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1]			
171	1.0	0.9999999962747097	1.0
[1, 0, 1]			
172	1.0	0.9999999962747097	1.0
[1, 0, 1]			
173	1.0	0.9999999962747097	1.0
[1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1]			
174	0.765625	0.9999999962747097	0.976562
[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1]			
175	0.998048	0.9999999962747097	0.999805
[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1]			
176	1.0	0.9999999962747097	1.0
[1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1]			



177	0.999939	0.9999999962747097	0.999957
[1, 0, 1]			
178	0.992142	0.9999999962747097	0.996857
[1, 0, 1]			
179	1.0	0.9999999962747097	1.0
[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1]			
180	1.0	0.9999999962747097	1.0
[1, 0, 1]			
181	0.999024	0.9999999962747097	0.999414
[1, 0, 1]			
182	0.998963	0.9999999962747097	0.999487
[1, 0, 1, 0, 1]			
183	0.998963	0.9999999962747097	0.999494
[1, 0, 1]			
184	0.999024	0.9999999962747097	0.999805
[1, 0, 1]			
185	1.0	0.9999999962747097	1.0
[1, 0, 1]			
186	1.0	0.9999999962747097	1.0
[1, 0, 1]			
187	1.0	0.9999999962747097	1.0
[1, 0, 1]			
188	1.0	0.9999999962747097	1.0
[1, 0, 1]			
189	0.999878	0.9999999962747097	0.999988
[1, 0, 1]			
190	1.0	0.9999999962747097	1.0
[1, 0, 1]			





Ciclo	Valor minimo	Valor maximo	Valor promedio	Cromosoma Máximo
1	0.105154	0.9685596821842396	0.648399	[1, 1, 1, 1, 1, 0, 1, 1, 0, 1, 1, 1, 0, 0, 0, 1, 1, 0, 0, 0, 1, 0, 0, 1, 0, 0, 1, 1, 1, 0]
2	0.532917	0.9647191441515435	0.803684	[1, 1, 1, 1, 1, 0, 1, 1, 0, 1, 1, 1, 0, 0, 0, 1, 1, 0, 0, 0, 1, 0, 0, 1, 0, 0, 1, 1, 1, 0]
3	0.532428	0.9647191441515435	0.804603	[1, 1, 1, 1, 1, 0, 1, 1, 0, 1, 1, 1, 0, 0, 0, 1, 1, 0, 0, 0, 1, 0, 0, 1, 0, 0, 1, 1, 1, 0]
4	0.532428	0.9647191459810357	0.810534	[1, 1, 1, 1, 1, 0, 1, 1, 0, 1, 1, 1, 0, 0, 0, 1, 1, 0, 0, 0, 1, 0, 0, 1, 0, 0, 1, 1, 1, 1]
5	0.532428	0.9647191459810357	0.853625	[1, 1, 1, 1, 1, 0, 1, 1, 0, 1, 1, 1, 0, 0, 0, 1, 1, 0, 0, 0, 1, 0, 0, 1, 0, 0, 1, 1, 1, 1]
6	0.532428	0.9647191459810357	0.896815	[1, 1, 1, 1, 1, 0, 1, 1, 0, 1, 1, 1, 0, 0, 0, 1, 1, 0, 0, 0, 1, 0, 0, 1, 0, 0, 1, 1, 1, 1]
7	0.96424	0.9685596821842396	0.964967	[1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 0, 0, 0, 1, 1, 0, 0, 0, 1, 0, 0, 1, 0, 0, 1, 1, 1, 0]
8	0.964285	0.9685596821842396	0.965442	[1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 0, 0, 0, 1, 1, 0, 0, 0, 1, 0, 0, 1, 0, 0, 1, 1, 1, 0]
9	0.964285	0.9685596821842396	0.965826	[1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 0, 0, 0, 1, 1, 0, 0, 0, 1, 0, 0, 1, 0, 0, 1, 1, 1, 0]

10	0.964285	0.9685596821842396	0.965784
	[1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 1, 1, 0, 0, 0, 1, 0, 0, 1, 0, 0, 1, 1, 1, 0]		
11	0.9643	0.9685596821842396	0.96617
	[1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 1, 1, 0, 0, 0, 1, 0, 0, 1, 0, 0, 1, 1, 1, 0]		
12	0.734794	0.9685596821842396	0.943603
	[1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 1, 1, 0, 0, 0, 1, 0, 0, 1, 0, 0, 1, 1, 1, 0]		
13	0.734794	0.9685596821842396	0.943989
	[1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 1, 1, 0, 0, 0, 1, 0, 0, 1, 0, 0, 1, 1, 1, 0]		
14	0.9643	0.968799969131024	0.966964
	[1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 1, 0, 0, 1, 1, 0, 0, 0, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1]		
15	0.9643	0.968799969131024	0.966988
	[1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 1, 0, 0, 1, 1, 0, 0, 0, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1]		
16	0.9643	0.9688002038008116	0.967102
	[1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 1, 0, 0, 1, 1, 0, 0, 0, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1]		
17	0.9643	0.9688002056341694	0.96835
	[1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 1, 0, 0, 1, 1, 0, 0, 0, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1]		
18	0.9688	0.9688006749738303	0.9688
	[1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 1, 0, 0, 1, 1, 0, 0, 0, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1]		
19	0.9688	0.9688006749738303	0.968801
	[1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 1, 0, 0, 1, 1, 0, 0, 0, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1]		
20	0.9688	0.9688006749738303	0.968801
	[1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 1, 0, 0, 1, 1, 0, 0, 0, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1]		
21	0.9688	0.9688006749738303	0.968801
	[1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 1, 0, 0, 1, 1, 0, 0, 0, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1]		
22	0.9688	0.9688006749738303	0.968801
	[1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 1, 0, 0, 1, 1, 0, 0, 0, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1]		
23	0.9688	0.9688006749738303	0.968801
	[1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 1, 0, 0, 1, 1, 0, 0, 0, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1]		

24	0.9688	0.9688006749738303	0.968801
[1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 1, 0, 0, 1, 1, 0, 0, 0, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1]			
25	0.9688	0.9688006749738303	0.968801
[1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 1, 0, 0, 1, 1, 0, 0, 0, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1]			
26	0.9688	0.9688006749738303	0.968801
[1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 1, 0, 0, 1, 1, 0, 0, 0, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1]			
27	0.968801	0.9688006749738303	0.968801
[1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 1, 0, 0, 1, 1, 0, 0, 0, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1]			
28	0.968801	0.9688081844238661	0.968801
[1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 0, 0, 1, 1, 1, 1, 0, 0, 0, 1, 1, 1]			
29	0.953482	0.9688081844238661	0.958079
[1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 0, 0, 1, 1, 1, 1, 0, 0, 0, 1, 1, 1]			
30	0.953482	0.9688081844238661	0.958079
[1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 0, 0, 1, 1, 1, 1, 0, 0, 0, 1, 1, 1]			
31	0.835331	0.9688081844238661	0.924166
[1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 0, 0, 1, 1, 1, 1, 0, 0, 0, 1, 1, 1]			
32	0.835331	0.9688081844238661	0.937513
[1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 0, 0, 1, 1, 1, 1, 0, 0, 0, 1, 1, 1]			
33	0.835331	0.9688081844238661	0.950862
[1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 0, 0, 1, 1, 1, 1, 0, 0, 0, 1, 1, 1]			
34	0.622465	0.9688081844238661	0.898007
[1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 0, 0, 1, 1, 1, 1, 0, 0, 0, 1, 1, 1]			
35	0.622465	0.9688081844238661	0.899539
[1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 0, 0, 1, 1, 1, 1, 0, 0, 0, 1, 1, 1]			
36	0.622471	0.9688081844238661	0.934173
[1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 0, 0, 1, 1, 1, 1, 0, 0, 0, 1, 1, 1]			
37	0.968801	0.9688081844238661	0.968807
[1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 0, 0, 1, 1, 1, 1, 0, 0, 0, 1, 1, 1]			

38	0.968801	0.9688081844238661	0.968807
[1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 0, 0, 1, 1, 1, 1, 0, 0, 0, 1, 1, 1]			
39	0.968568	0.9688081844238661	0.968781
[1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 0, 0, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1]			
40	0.968793	0.9688081844238661	0.968804
[1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 0, 0, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1]			
41	0.96832	0.9689283249147274	0.968771
[1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0, 1, 1, 0, 1, 1, 1, 0, 0, 1, 1, 1, 1, 0, 0, 0, 1, 1, 1]			
42	0.96832	0.9689283249147274	0.96877
[1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0, 1, 1, 0, 1, 1, 1, 0, 0, 1, 1, 1, 1, 0, 0, 0, 1, 1, 1]			
43	0.96832	0.9689283249147274	0.968771
[1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0, 1, 1, 0, 1, 1, 1, 0, 0, 1, 1, 1, 1, 0, 0, 0, 1, 1, 1]			
44	0.96832	0.9689283249147274	0.968771
[1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0, 1, 1, 0, 1, 1, 1, 0, 0, 1, 1, 1, 1, 0, 0, 0, 1, 1, 1]			
45	0.96832	0.9689283249147274	0.968687
[1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0, 1, 1, 0, 1, 1, 1, 0, 0, 1, 1, 1, 1, 0, 0, 0, 1, 1, 1]			
46	0.96832	0.9689283249147274	0.968733
[1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0, 1, 1, 0, 1, 1, 1, 0, 0, 1, 1, 1, 1, 0, 0, 0, 1, 1, 1]			
47	0.96832	0.9689283249147274	0.968807
[1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0, 1, 1, 0, 1, 1, 1, 0, 0, 1, 1, 1, 1, 0, 0, 0, 1, 1, 1]			
48	0.968801	0.9689283249147274	0.968879
[1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0, 1, 1, 0, 1, 1, 1, 0, 0, 1, 1, 1, 1, 0, 0, 0, 1, 1, 1]			
49	0.968448	0.9688006749738303	0.968518
[1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 0, 0, 1, 1, 0, 0, 0, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1]			
50	0.968448	0.9688006749738303	0.968554
[1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 0, 0, 1, 1, 0, 0, 0, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1]			
51	0.968448	0.9688006749738303	0.96866
[1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 0, 0, 1, 1, 0, 0, 0, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1]			

52	0.968448	0.9688006749738303	0.968695
[1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 0, 0, 1, 1, 0, 0, 0, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1]			
53	0.968448	0.9688006749738303	0.968624
[1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 0, 0, 1, 1, 0, 0, 0, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1]			
54	0.968448	0.9688006749738303	0.96873
[1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 0, 0, 1, 1, 0, 0, 0, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1]			
55	0.968448	0.9688007043075628	0.96873
[1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 0, 0, 1, 0, 0, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1]			
56	0.968801	0.9688908196302095	0.968873
[1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 0, 0, 1, 0, 0, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1]			
57	0.968801	0.9688908196302095	0.968855
[1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 0, 0, 1, 0, 0, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1]			
58	0.968801	0.9688908196302095	0.968855
[1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 0, 0, 1, 0, 0, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1]			
59	0.968801	0.9688908196302095	0.968873
[1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 0, 0, 1, 0, 0, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1]			
60	0.968801	0.9688983000943945	0.968874
[1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 0, 0, 1, 1, 0, 0, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1]			
61	0.968801	0.9688983000943945	0.968874
[1, 0, 1, 1, 1, 0, 1, 1, 1, 1, 0, 1, 1, 1, 0, 0, 1, 1, 0, 0, 1, 1, 1, 0, 1, 0, 1, 1, 1, 1]			
62	0.968801	0.9688983000943945	0.968886
[1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 0, 0, 1, 1, 0, 0, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1]			
63	0.968891	0.9688983587648159	0.968895
[1, 0, 1, 1, 1, 0, 1, 1, 1, 1, 0, 1, 1, 1, 0, 0, 1, 1, 0, 0, 1, 1, 1, 0, 1, 0, 1, 1, 1, 1]			
64	0.968891	0.9688983587648159	0.968896
[1, 0, 1, 1, 1, 0, 1, 1, 1, 1, 0, 1, 1, 1, 0, 0, 1, 1, 0, 0, 1, 1, 1, 0, 1, 0, 1, 1, 1, 1]			
65	0.968891	0.9688983587648159	0.968897
[1, 0, 1, 1, 1, 0, 1, 1, 1, 1, 0, 1, 1, 1, 0, 0, 1, 1, 0, 0, 1, 1, 1, 0, 1, 0, 1, 1, 1, 1]			

66	0.968891	0.9688983587648159	0.968898
[1, 0, 1, 1, 1, 0, 1, 1, 1, 1, 0, 1, 1, 1, 0, 0, 1, 1, 0, 0, 1, 1, 1, 0, 1, 0, 1, 1, 1, 1]			
67	0.968891	0.9688983587648159	0.968898
[1, 0, 1, 1, 1, 0, 1, 1, 1, 1, 0, 1, 1, 1, 0, 0, 1, 1, 0, 0, 1, 1, 1, 0, 1, 0, 1, 1, 1, 1]			
68	0.967937	0.9688983587648159	0.968418
[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 0, 1, 1, 1, 0, 0, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1]			
69	0.848864	0.9688983587648159	0.956414
[1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 1, 1, 0, 0, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1]			
70	0.848864	0.9688983587648159	0.956607
[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 0, 1, 1, 1, 0, 0, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1]			
71	0.967937	0.9688983587648159	0.968418
[1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 1, 1, 0, 0, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1]			
72	0.538518	0.9688983587648159	0.88263
[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 0, 1, 1, 1, 0, 0, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1]			
73	0.538518	0.9688983587648159	0.925668
[1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 1, 1, 0, 0, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1]			
74	0.967937	0.9688983587648159	0.968706
[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 0, 1, 1, 1, 0, 0, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1]			
75	0.967937	0.9688983587648159	0.968802
[1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 1, 1, 0, 0, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1]			
76	0.967937	0.9688983587648159	0.968802
[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 0, 1, 1, 1, 0, 0, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1]			
77	0.968898	0.9688983587648159	0.968898
[1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 1, 1, 0, 0, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1]			
78	0.968898	0.9688983587648159	0.968898
[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 0, 1, 1, 1, 0, 0, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1]			
79	0.968898	0.9689283982538908	0.968907
[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 0, 1, 1, 1, 0, 0, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1]			



80	0.968898	0.9689283982538908	0.968913
[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 0, 1, 1, 1, 0, 0, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1]			
81	0.968898	0.9689283982538908	0.968916
[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 0, 1, 1, 1, 0, 0, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1]			
82	0.968898	0.9689283982538908	0.968916
[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 0, 1, 1, 1, 0, 0, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1]			
83	0.968898	0.9689283982538908	0.968922
[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 0, 1, 1, 1, 0, 0, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1]			
84	0.968688	0.9689283982538908	0.968733
[1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1]			
85	0.968688	0.9689283982538908	0.968757
[1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1]			
86	0.968688	0.9689283982538908	0.968799
[1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1]			
87	0.968688	0.9689283982538908	0.968799
[1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1]			
88	0.968898	0.9996890965392373	0.987379
[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 0, 1, 1, 1, 0, 0, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1]			
89	0.968898	0.9996890965392373	0.993534
[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 0, 1, 1, 1, 0, 0, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1]			
90	0.968928	0.9996890965392373	0.993537
[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 0, 1, 1, 1, 0, 0, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1]			
91	0.968928	0.9997501231381085	0.996619
[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1]			
92	0.999689	0.9997501231381085	0.999707
[1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1]			
93	0.999689	0.9997501231381085	0.999726
[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1]			

94	0.999689	0.9997501231381085	0.99972
[1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1]			
95	0.999689	0.9997501231381085	0.999732
[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1]			
96	0.562313	0.9997501231381085	0.912263
[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1]			
97	0.562313	0.9997501231381085	0.956006
[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1]			
98	0.99975	0.9997501231381085	0.99975
[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1]			
99	0.99975	0.9997501231381085	0.99975
[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1]			
100	0.765406	0.9997501231381085	0.976316
[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1]			

El cromosoma con el valor más alto es:

[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1]

## Conclusion:

En elitismo, las poblaciones convergen más rápidamente hacia un valor máximo cercano a la solución óptima.

Cuando cambiamos y aumentamos la probabilidad de mutación, se observó cambios erráticos en la gráficas de los valores y, mientras más aumentamos la probabilidad, más difícil era que converjan los valores de la población.

A mayor cantidad de ciclos, mayor probabilidad de que la población tenga valores máximos y promedios mayores, aun así vemos en varios casos que solo hacen falta 20 generaciones para llegar a valores altos aunque al aumentar el número a 100 o a 200 nos aseguramos de que existen varias generaciones con valores altos.

Creamos varias funciones vistas en el apunte como por ejemplo las relacionadas a “máscaras”, pero al ser aplicadas en un universo no dan buenos resultados que ayuden a los valores máximos y modifican severamente los valores promedio.