

# **Trabajo Práctico N° 1**

## **Algoritmos Genéticos**

### **2020**

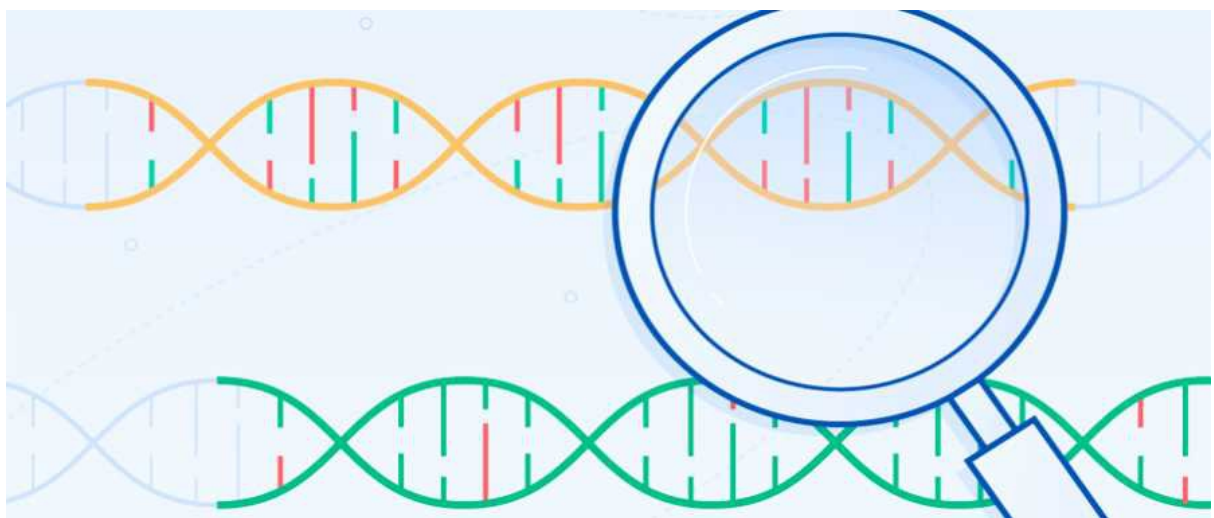
## **Integrantes**

**Luciano Vannelli 45744 - [lucianovannelli@gmail.com](mailto:lucianovannelli@gmail.com)**

**Manuel Bahamonde - 45699 - [manuelbahamonde6@gmail.com](mailto:manuelbahamonde6@gmail.com)**

**Franco Ceschan - 42737 - [francoceschan@gmail.com](mailto:francoceschan@gmail.com)**

**Milena Lebedinsky - 43982 - [milenalebedin@gmail.com](mailto:milenalebedin@gmail.com)**



# Índice

Enunciado.....	2
Desarrollo.....	4
Metodología de desarrollo abordada.....	4
Herramientas de programación utilizadas.....	5
Forma de trabajo en equipo.....	5
El programa.....	6
Código del programa.....	6
Salida.....	12
Gráficas.....	13
Conclusiones.....	31

## Enunciado

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

la función:

$f(x) = (x/\text{coef})^2$  en el dominio  $[0, 230 - 1]$

donde coef = 230 -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 elite 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 basada 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 posean 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.

## Desarrollo

### Metodología de desarrollo abordada

#### **Clase población**

Guarda en sus atributos información sobre los resultados de las corridas.

#### **Clase cromosoma**

Guarda en sus atributos información sobre los cromosomas.

#### **Generación de la población inicial:**

Consta de 3 funciones que se llaman entre ellas de forma iterativa para generar los genes y cromosomas de la población inicial

#### **Guardado de datos:**

Esta función la dividimos en el **cálculo de los datos de los cromosomas** y luego el **seteo de los datos de la población y el fitness**. Al crear cada cromosoma, en el mismo constructor de la clase, calculamos el valor que este tiene y lo evaluamos en la función objetivo, guardando ambos datos. Luego, en el seteo de los datos de la población, iteramos sobre todos los cromosomas de la misma para encontrar:

- El que tiene mayor y menor función objetivo, guardando tanto el valor como el cromosoma en un objeto de la clase Cromosoma
- La suma de los valores que cada cromosoma obtenía en la función objetivo
- El valor promedio de los valores de función objetivo de toda la población
- El fitness de cada cromosoma (es el único valor que no podemos calcular en el constructor del mismo ya que necesitamos primero calcular el promedio de función objetivo)

#### **Método de selección**

En este método realizamos la selección con el método de ruleta: creamos un array de pares de cromosomas y vamos agregando elementos que contienen dos cromosomas los cuales elegimos aleatoriamente teniendo en cuenta el fitness. Luego los devolvemos para la siguiente etapa.

#### **Crossover y mutación**

Contiene una función que se encarga de realizar tanto el crossover como la mutación en

caso de corresponder.

### **Mensaje final con datos de la generación:**

Incluimos una función que muestra los datos finales de la generación que acaba de finalizar.

### **Helpers:**

Dentro de esta región incluimos funciones que realizan tareas relacionadas con la forma de presentar la información e interactuar con el usuario y no con el AG en sí.

Al correr el programa nos pregunta primero si queremos realizar las corridas con elitismo o no, luego de las corridas muestra un gráfico informativo y al cerrar este se abre un archivo excel con la información más detallada.

## Herramientas de programación utilizadas

**Lenguaje de programación** : Python

### **Librerías utilizadas:**

**random** : Ofrece generadores de números aleatorios que utilizamos para múltiples funciones del programa.

**numpy** : Agrega soporte para vectores y matrices, constituye una biblioteca de funciones matemáticas de alto nivel para operar con ellos.

**xlsxwriter** : Utilizado para editar archivos Excel.

**operator** : Ofrece herramientas útiles para hacer referencia con facilidad a datos organizados en listas y atributos de clases.

**os** : Permite realizar operaciones dependiente del Sistema Operativo, como crear una carpeta, finalizar un proceso, etc. La utilizamos para crear y abrir archivos de excel.

**pandas**: es una biblioteca para manipulación y análisis de datos. La utilizamos para convertir el archivo de excel en un archivo de valores separados por comas (csv).

**matplotlib**: es una biblioteca para la generación de gráficos a partir de datos contenidos en listas o arrays.

## Forma de trabajo en equipo

Para afrontar la virtualidad, nos manejamos con aplicaciones de mensajería, de chat de voz, con aplicaciones de almacenamiento en la nube y con una extensión para el IDE que utilizamos para poder editar y ver el código todos en tiempo real.

- **Visual Studio Code**: el IDE con el que escribimos el algoritmo

- **Extensión Live Share:** la extensión del IDE que permitió que todos podamos editar y ver el código en tiempo real
- **Repositorios en Google Drive:** lo utilizamos para guardar el código del algoritmo actualizado
- **Grupo de Whatsapp:** utilizado para coordinar reuniones y notificar cambios y modificaciones realizadas en el código.
- **Discord:** lo utilizamos para discutir el código que escribíamos por voz.

## El programa

### Código del programa

```
import random
import time
import numpy
import xlswriter
from operator import attrgetter
import os
import pandas as pd
import matplotlib.pyplot as plt

"EJERCICIO: Método de Selección: Ruleta. Método de Crossover: 1 Punto.
Método de Mutación: invertida."

"Datos del algoritmo genetico (para distintos ejercicios, a menos que
cambie el metodo de crossover y mutacion, solo hay que modificar estos
datos y f(x))"

pCrossover = 0.75
pMutacion = 0.05
cantIndividuos = 10
cantGenes = 30
ciclos = 200
tamanoElite = 2
nombreExcel = 'ag.xlsx'

"Variables globales"

class Poblacion:
    def __init__(self, cromosomas):
        self.cromosomas = cromosomas

        self.cromMaxCiclo = Cromosoma() #Inicializamos el cromosoma
maximo del ciclo un cromosoma con objetivo 0 (para fxMax usamos el
objeto del cromosoma maximo)
```

```

        self.fxMin = 100
        self.fxSum = 0
        self.fxProm = 0

#Etapa 2: Guardado de datos
def calcularYSetearDatos(self):
    for cromosoma in self.cromosomas:

        #Calculo del fxMax
        if(cromosoma.fx > self.cromMaxCiclo.fx):
            self.cromMaxCiclo = cromosoma

        #Calculo del fxMin
        if(cromosoma.fx < self.fxMin):
            self.fxMin = cromosoma.fx

        #Calculo de la suma
        self.fxSum += cromosoma.fx

    #Calculo del promedio
    self.fxProm = (self.fxSum/len(self.cromosomas))

    #Calculo del fitness de cada cromosoma
    for cromosoma in self.cromosomas:
        cromosoma.calcularFitness(self.fxSum)

#Etapa 3: Seleccion
def seleccionRuleta(self,elitismo):
    individuos = cantIndividuos
    if(elitismo):
        individuos -= 2

    pares = []
    for i in range(int(individuos/2)):
        pares.append([])
        for j in range(2):

pares[i].append(random.choices(self.cromosomas,self.getListaFtn())[0])

        return pares

#Etapa 4: Crossover y mutación
def aplicarCrossoverMutacion(self,pares):

```

```

        #Aplica tanto el crossover como la mutacion (solo en caso de
que hayan sido aprobados)
        sigPoblacion = []
        for par in pares:
            if(self.realizar(pCrossover)):
                puntoCorte = random.randint(0,cantGenes)
            else:
                puntoCorte = cantGenes

            for i in range(2):
                bits = []
                for j in range(puntoCorte):
                    bits.append(par[i].bits[j])
                for j in range(puntoCorte,cantGenes):
                    bits.append(par[abs(i-1)].bits[j])

                if(self.realizar(pMutacion)):
                    puntoAMutar = random.randint(0,cantGenes-1)
                    bits[puntoAMutar] = abs(bits[puntoAMutar]-1)

                cromosoma = Cromosoma(bits)

                sigPoblacion.append(cromosoma)
        return sigPoblacion

    def realizar(self,probabilidadTrue):
        #Recibe una probabilidad para realizar algo y devuelve True si
se debe realizar o False si no (usado para decidir crossover y
mutacion)

        return (random.choices([0,1],[1-
probabilidadTrue,probabilidadTrue])[0] == 1)

    #Helpers
    def aplicarElitismo(self,tamanoElite):
        cromosomasElite = []
        for i in range(tamanoElite):
            unMaximo = max(self.cromosomas,key=attrgetter('ftn'))
#Sacamos el cromosoma con mayor fitness

            #Lo eliminamos de los cromosomas de la pobl actual y a la
vez lo asignamos a la lista de elite (pop elimina el objeto y a su vez
lo devuelve)

            cromosomasElite.append(self.cromosomas.pop(self.cromosomas.index(unMaxi

```



```

mo)))

        return cromosomasElite

def getListaFtn(self):
    listaFtn = []
    for cromosoma in self.cromosomas:
        listaFtn.append(cromosoma.ftn)

class Cromosoma:
    def __init__(self, bits=None):
        #Para el cromosoma maximo inicial, lo inicializamos con bits =
        #None, en ese caso lo unico que nos interesa es el Fx para luego poder
        #compararlo
        if(bits != None):
            self.bits = bits
            self.x = (int(self.getBinario(),2)) #int(string,2=base
            #binario)
            self.fx = self.f()
            self.ftn = 0
        else:
            self.fx = 0

    def calcularFitness(self, sumaFx):
        self.ftn = (self.fx/sumaFx)

    def f(self):
        #Evalua la x del cromosoma en una funcion
        return (self.x/(pow(2,30) - 1))

    def getBinario(self):
        #Concatena en un string todos los elementos del cromosoma
        #genes) y los devuelve
        binario = ""
        for gen in self.bits:
            binario += str(gen)

        return binario

def empezar():
    global poblacion, cromosomaMaximoCorrida
    #Inicializaciones
    inicializarExcel() #Para tabla y graficos
    cromosomaMaximoCorrida = Cromosoma() #Inicializamos el cromosoma

```

```

maximo como un cromosoma con objetivo 0
    elitismo = preguntarElitismo()
    poblacion = Poblacion(getPoblacionInicial())

    print("Comenzando algoritmo con "+str(ciclos)+" ciclos
solicitados")

    for i in range(ciclos):
        print()
        print()
        sigCromosomas = [] #Lista de cromosomas siguientes (utilizados
cuando se activa elitismo)
        poblacion.calcularYSetearDatos()

        #Actualizacion del cromosoma maximo de la corrida
        if(poblacion.cromMaxCiclo.fx > cromosomaMaximoCorrida.fx):
            cromosomaMaximoCorrida = poblacion.cromMaxCiclo

        if(elitismo):
            #Sacamos los cromosomas con mejor fitness
            sigCromosomas = poblacion.aplicarElitismo(tamanoElite)

        pares = poblacion.seleccionRuleta(elitismo)

        #Final del ciclo actual
        agregarDatos(i+1)
        finCiclo(i+1)

        poblacion =
Poblacion(poblacion.aplicarCrossoverMutacion(pares))
        poblacion.cromosomas += sigCromosomas #Si hay elitismo,
sigCromosomas va a tener cromosomas y aca se agregan, sino va a estar
vacio y no se va a agregar nada

        workbook.close()
        agregaGrafico()
        os.system(nombreExcel) #Abrimos el archivo recién creado

#Etapa 1: Generación de la población inicial
def getPoblacionInicial():
    cromosomas = []
    for bits in generarPoblacionInicial():
        cromosomas.append(Cromosoma(bits))

```

```

    return cromosomas
def generarPoblacionInicial():
    return [generarCromosomaInicial() for i in range(cantIndividuos)]
def generarCromosomaInicial():
    return [random.randint(0, 1) for i in range(cantGenes)]

#Etapa 5: Mensaje final con los datos de la generacion
def finCiclo(numCiclo):
    #Muestra los datos finales de la generacion que acaba de finalizar
    print("CICLO "+str(numCiclo)+" COMPLETADO. Datos del mismo:")
    print("FX --> SUMA = "+str(poblacion.fxSum)+" - VALOR MAXIMO = "+str(poblacion.cromMaxCiclo.fx)+" - VALOR MINIMO = "+str(poblacion.fxMin)+" - VALOR PROMEDIO = "+str(poblacion.fxProm))
    print("CROMOSOMA MAXIMO --> "+str(poblacion.cromMaxCiclo.bits))

    if(numCiclo == ciclos):
        print("Algoritmo finalizado: se cumplieron los ciclos solicitados")
        print("Cromosoma maximo de toda la corrida: "+str(cromosomaMaximoCorrida.bits))
        print("Funcion objetivo del cromosoma maximo de toda la corrida: "+str(cromosomaMaximoCorrida.fx))

#Helpers
def esperarEntrada():
    #Pide un input para continuar la ejecucion del algoritmo
    print("Presione enter para continuar")
    input()
def ingresarOpcion(min,max):
    resp = -1
    while(resp < min or resp > max):
        try:
            resp = int(input())
        except:
            print("Entrada no valida")
            resp = -1
    return resp
def agregaGrafico():
    excel=pd.read_excel(nombreExcel)
    excel.to_csv('ag.csv', index=None, header=True)

    datos=pd.read_csv('ag.csv',header=0)

```

```

fix, ax = plt.subplots()
ax.plot(datos['Minimo'], label='Minimo')
ax.plot(datos['Maximo'], label='Maximo')
ax.plot(datos['Promedio'], label='Promedio')
plt.legend(loc='lower right')

plt.show()
def agregarDatos(i):
    cont = i

    worksheet.write(cont,0,i)
    worksheet.write(cont,1,poblacion.fxMin)
    worksheet.write(cont,2,poblacion.cromMaxCiclo.fx)
    worksheet.write(cont,3,poblacion.cromMaxCiclo.getBinario())
    worksheet.write(cont,4,poblacion.fxProm)
def inicializarExcel():
    global worksheet,chart,workbook

    os.popen('del '+nombreExcel) #A veces se da que cuando ya hay un
archivo con el mismo nombre del que vamos a crear, no se crea. Entonces
lo eliminamos
    workbook = xlswriter.Workbook(nombreExcel)
    worksheet = workbook.add_worksheet()
    chart = workbook.add_chart({'type': 'line'})
    worksheet.write(0,0, 'Ciclo')
    worksheet.write(0,1, 'Minimo')
    worksheet.write(0,2, 'Maximo')
    worksheet.write(0,3, 'Cromosoma maximo')
    worksheet.write(0,4, 'Promedio')

def preguntarElitismo():
    print("Desea incluir elitismo en el algoritmo? (0 no, 1 si)")
    return (ingresarOpcion(0,1) == 1)

empezar()

```

## Salida

Se muestra, para cada generación, el cromosoma máximo y su valor, el valor del cromosoma mínimo y el valor promedio.

```

CICLO 134 COMPLETADO. Datos del mismo:
FX --> SUMA = 4.579210836979757 - VALOR MAXIMO = 0.9089890140192481 - VALOR MINIMO = 0.20167546458698388 - VALOR PROMEDIO = 0.45792108369797574
CROMOSOMA MAXIMO --> [1, 1, 1, 0, 1, 0, 0, 0, 1, 0, 1, 1, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1]

```

```

CICLO 200 COMPLETADO. Datos del mismo:
FX --> SUMA = 3.224133589513724 - VALOR MAXIMO = 0.9089890140192481 - VALOR MINIMO = 0.20167594887472312 - VALOR PROMEDIO = 0.3224133589513724
CROMOSOMA MAXIMO --> [1, 1, 1, 0, 1, 0, 0, 0, 1, 0, 1, 1, 0, 0, 1, 1, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1]

```

Al finalizar todos los ciclos se muestra el cromosoma máximo de toda la corrida, y su correspondiente valor para la función objetivo.

```

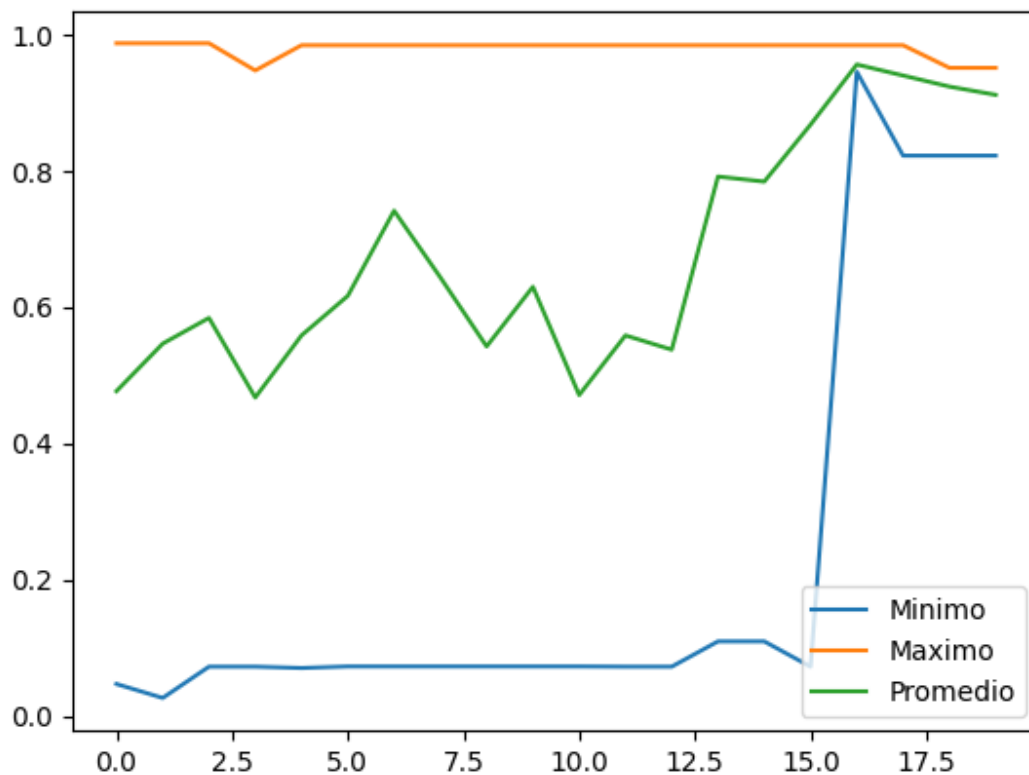
Algoritmo finalizado: se cumplieron los ciclos solicitados
Cromosoma maximo de toda la corrida: [1, 1, 1, 0, 1, 0, 0, 0, 1, 0, 1, 1, 0, 0, 1, 1, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1]
Funcion objetivo del cromosoma maximo de toda la corrida: 0.9089890140192481
El sistema no puede ejecutar el programa especificado.

```

Se obtienen un excel y un gráfico, generados a partir de estos datos.

## Gráficas

### 20 ciclos sin elitismo



```

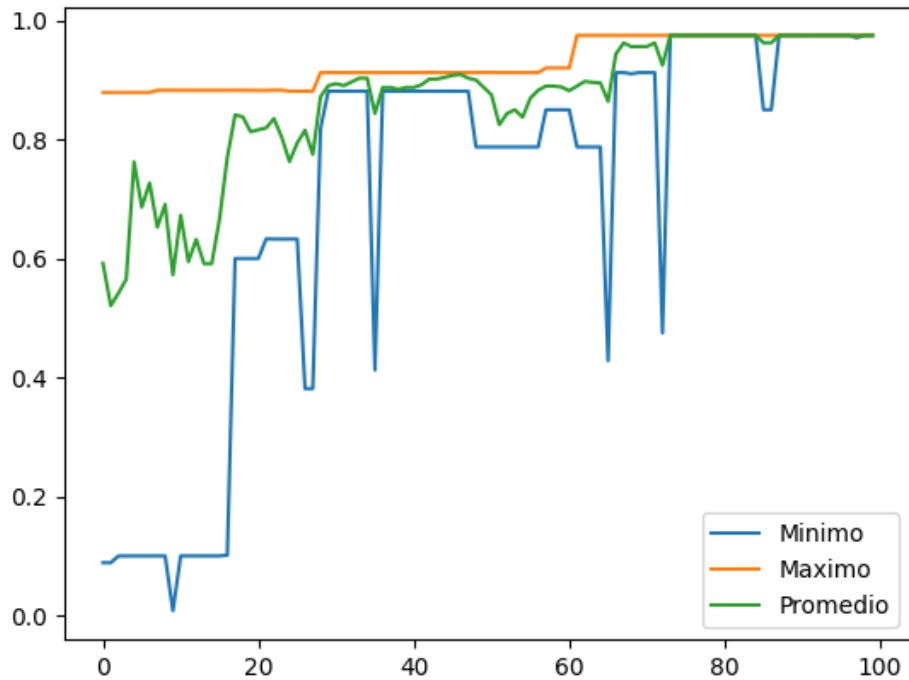
Algoritmo finalizado: se cumplieron los ciclos solicitados
Cromosoma maximo de toda la corrida: [1, 1, 1, 1, 1, 1, 0, 0, 1, 1, 0, 1, 1, 0, 0, 1, 0, 0, 1, 1, 1, 1, 1, 0, 0, 0, 1, 1, 1]
Funcion objetivo del cromosoma maximo de toda la corrida: 0.9876899197582993

```

Ciclo	Mínimo	Máximo	Cromosoma máximo	Promedio
1	0,047233899167957	0,987689812656203	111111001101100100111101010100	0,477033292853305
2	0,026727993066169	0,987689919758299	111111001101100100111111000111	0,546741380958577
3	0,072646084309226	0,987689919758299	111111001101100100111111000111	0,584491940200787
4	0,072646084309226	0,947646085124133	111100101001100011101111000011	0,467697224735932

5	0,070697731404284	0,984760232255571	1111110000011001001111110001 11	0,558876564315405
6	0,072966869057125	0,984760232255571	1111110000011001001111110001 11	0,616883136720288
7	0,072966869057125	0,984760232255571	1111110000011001001111110001 11	0,741820505766031
8	0,072966869057125	0,984755460158694	1111110000011000111011110000 11	0,643968180235446
9	0,072966816903061	0,984763089553232	1111110000011001011011110000 11	0,54266619825984
10	0,072966995716996	0,984755460158694	1111110000011000111011110000 11	0,630134119866541
11	0,072966995716996	0,984755460158694	1111110000011000111011110000 11	0,471346131219851
12	0,072646084309226	0,984755460158694	1111110000011000111011110000 11	0,558814040160565
13	0,072646084309226	0,984755460158694	1111110000011000111011110000 11	0,537914434390063
14	0,109763088738325	0,984751645461425	1111110000011000101011110000 11	0,791983929827794
15	0,109763088738325	0,984751645461425	1111110000011000101011110000 11	0,784562817760336
16	0,072653713703764	0,984763089553232	1111110000011001011011110000 11	0,868156567838189
17	0,945700589516853	0,984763089553232	1111110000011001011011110000 11	0,956437819410523
18	0,822653714402256	0,984763089553232	1111110000011001011011110000 11	0,940227643344764
19	0,822653714402256	0,951552335127771	1111001110011000111011110000 11	0,924017850238847
20	0,822653714402256	0,951552335127771	1111001110011000111011110000 11	0,911907712288115

## 100 ciclos sin elitismo



Algoritmo finalizado: se cumplieron los ciclos solicitados  
 Cromosoma maximo de toda la corrida: [1, 1, 1, 1, 1, 0, 0, 1, 1, 0, 0, 0, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 0, 1, 0, 1, 0, 1]  
 Funcion objetivo del cromosoma maximo de toda la corrida: 0.9748515691373978

Ciclo	Mínimo	Máximo	Cromosoma máximo	Promedio
1	0,08846389231129	0,879017141534963	11100001000001110100010001110 0	0,591642463385726
2	0,088465453207926	0,879017141534963	11100001000001110100010001110 0	0,520709990636176
3	0,099754323344449	0,879017141534963	11100001000001110100010001110 0	0,541583313645407
4	0,09975078338734	0,879020681492072	1110000100000111011111111010 1	0,565177269992584
5	0,099846167582875	0,879018774143438	1110000100000111010111111010 1	0,762250227166573
6	0,099846167582875	0,879018774143438	1110000100000111010111111010 1	0,686046574717431
7	0,099846166651553	0,87901877507476	1110000100000111010111111011 0	0,726826480428527
8	0,099846352916068	0,88264360919841	11100001111101001110111001111 0	0,65255144727654
9	0,099848656076797	0,882765679511023	1110000111111001110111001111 0	0,690788639980171
10	0,007643608383502	0,882765700931443	111000011111100111011101010 1	0,572436749816348
11	0,099848737101861	0,882641315350906	11100001111101001100011111111 1	0,6724273609672
12	0,099848656076797	0,882674126776563	11100001111101101110111001111 0	0,594489893498355

13	0,099848730582603	0,882674126776563	11100001111101101110011110	0,631948978483629
14	0,099848730582603	0,882673164720343	11100001111101101111001010	0,591166438899158
15	0,099848730582603	0,882681756171101	11100001111101101110011110	0,591190282340339
16	0,099848737101861	0,882681756171101	11100001111101101110011110	0,666189423823906
17	0,10117902336752	0,882681756171101	11100001111101101110011110	0,769423606218233
18	0,59983625039443	0,882681756171101	11100001111101101110011110	0,84116575190906
19	0,599836303479817	0,882687469835102	111000011111011110011001010	0,837931562343548
20	0,599836303479817	0,882444282884192	1110000111100111110111001010	0,812932515156393
21	0,5998353498055	0,882444282884192	1110000111100111110111001010	0,816193587254913
22	0,632810493589202	0,882444281952869	1110000111100111110111001010	0,819356728921976
23	0,632444281720039	0,882810493822033	111000011111111110111001010	0,834810915249224
24	0,632444284514006	0,882810493822033	111000011111111110111001010	0,803731535006065
25	0,632437605999818	0,881101518758667	1110000110001111101111001111	0,762875275372411
26	0,632437605999818	0,881101518758667	1110000110001111101111001111	0,794003206020224
27	0,381101510842425	0,881101518758667	1110000110001111101111001111	0,815476333550621
28	0,38110150897978	0,881101518758667	1110000110001111101111001111	0,774851465760592
29	0,818601509387234	0,912351509474545	1110100110001111101111001010	0,871726511671884
30	0,881101509445441	0,912351509474545	1110100110001111101111001010	0,890476509826702
31	0,881101509445441	0,912351509474545	1110100110001111101111001010	0,893601510015876
32	0,881101390236152	0,912351509474545	1110100110001111101111001010	0,890476497905772
33	0,881097694748172	0,91235151133719	1110100110001111101111001011	0,896726128362795
34	0,881101509445441	0,91235151133719	1110100110001111101111001011	0,902976509838343
35	0,881101509445441	0,91235151133719	1110100110001111101111001011	0,902976128554881
36	0,412351509008884	0,912351509474545	1110100110001111101111001010	0,843601104285196
37	0,881101271026862	0,912351509474545	1110100110001111101111001010	0,887351056642226
38	0,881101271026862	0,912351509474545	1110100110001111101111001010	0,88735139098703
39	0,881101271026862	0,912351509474545	1110100110001111101111001010	0,884226318713488

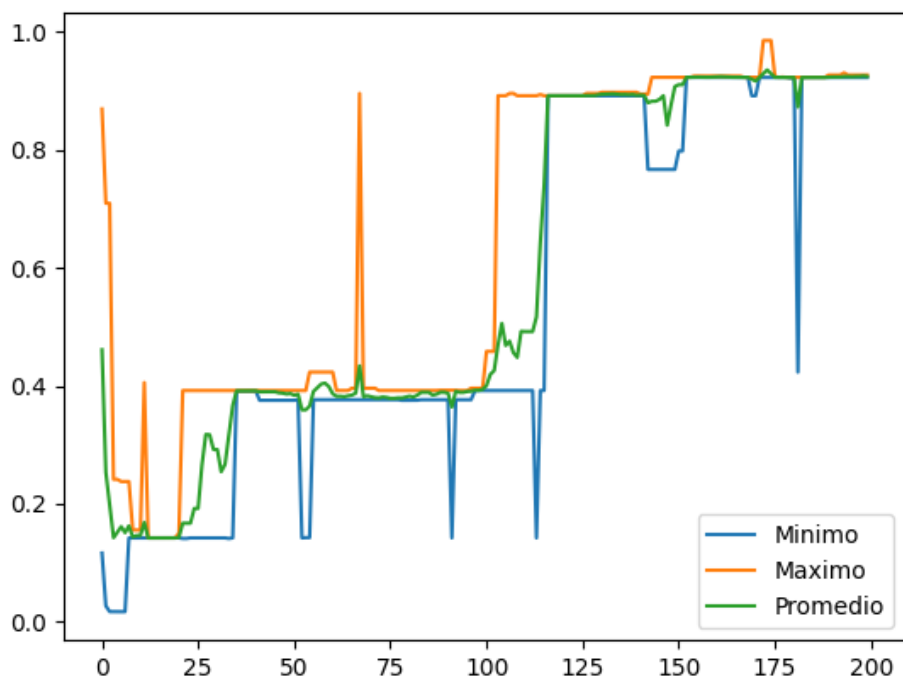


40	0,881101271026862	0,912351271055966	1110100110001111101101001010 1	0,887351342558257
41	0,881101271026862	0,912351509474545	1110100110001111101111001010 1	0,887351342558257
42	0,881101271026862	0,912351509474545	1110100110001111101111001010 1	0,892038818720764
43	0,881101271026862	0,912595411681193	1110100110011111101101001010 1	0,901438256633876
44	0,881101271026862	0,912351509474545	1110100110001111101111001010 1	0,901413723362064
45	0,881101271026862	0,912351509474545	1110100110001111101111001010 1	0,904538842574264
46	0,881101271026862	0,912351509474545	1110100110001111101111001010 1	0,907663484949305
47	0,881097694748172	0,912351509474545	1110100110001111101111001010 1	0,909226056476334
48	0,881097694748172	0,912351509474545	1110100110001111101111001010 1	0,90297529353106
49	0,787351270939551	0,912351509474545	1110100110001111101111001010 1	0,899851366411756
50	0,787351270939551	0,912595411681193	1110100110011111101101001010 1	0,88737575662078
51	0,787351270939551	0,912595411681193	1110100110011111101101001010 1	0,874875708925422
52	0,787351270939551	0,912351509474545	1110100110001111101111001010 1	0,824875708878856
53	0,787351256038389	0,912351509474545	1110100110001111101111001010 1	0,843625755089918
54	0,78735150935813	0,912351509474545	1110100110001111101111001010 1	0,84987585046317
55	0,78735150935813	0,912351271055966	1110100110001111101101001010 1	0,837400264514052
56	0,787351270939551	0,912351509474545	1110100110001111101111001010 1	0,869407028955787
57	0,787351270939551	0,912351509474545	1110100110001111101111001010 1	0,882662339026725
58	0,849851270997758	0,920163771063242	1110101110001111101101001010 1	0,889695089580207
59	0,849851270997758	0,920163771063242	1110101110001111101101001010 1	0,889695089580207
60	0,849851256096597	0,920164009481821	1110101110001111101111001010 1	0,888131899654988
61	0,849851256096597	0,920164009481821	1110101110001111101111001010 1	0,881881947332883
62	0,787343641545012	0,974851509532753	1111100110001111101111001010 1	0,889694472672133
63	0,787343879963592	0,974851509532753	1111100110001111101111001010 1	0,897506972679409
64	0,787343879963592	0,974851509532753	1111100110001111101111001010 1	0,895944472677954
65	0,787343879963592	0,974851509532753	1111100110001111101111001010 1	0,895163223422284
66	0,427976270604856	0,974851509532753	1111100110001111101111001010 1	0,863913079596975

67	0,912343880080007	0,974851509532753	1111100110001111101111001010 1	0,943599220685288
68	0,912343880080007	0,974851509532753	1111100110001111101111001010 1	0,962153145263114
69	0,910398384472726	0,974851509532753	1111100110001111101111001010 1	0,955904671136201
70	0,912343880080007	0,974851509532753	1111100110001111101111001010 1	0,956099983636383
71	0,912343880080007	0,974851509532753	1111100110001111101111001010 1	0,956099983636383
72	0,912343880080007	0,974851509532753	1111100110001111101111001010 1	0,962350746488525
73	0,474851508135769	0,974851509532753	1111100110001111101111001010 1	0,924851509393054
74	0,974851509532753	0,974851569137398	1111100110001111101111101010 1	0,974851515493217
75	0,974851509532753	0,974851509532753	1111100110001111101111001010 1	0,974851509532753
76	0,974851509532753	0,974851539335075	1111100110001111101111011010 1	0,974851512512985
77	0,974851509532753	0,974851539335075	1111100110001111101111011010 1	0,974851512512985
78	0,974851509532753	0,974851539335075	1111100110001111101111011010 1	0,974851512512985
79	0,974851271114174	0,974851509532753	1111100110001111101111001010 1	0,974851485690895
80	0,974851271114174	0,974851509532753	1111100110001111101111001010 1	0,974851461849037
81	0,974851271114174	0,974851509532753	1111100110001111101111001010 1	0,974851438007179
82	0,974851271114174	0,974851509532753	1111100110001111101111001010 1	0,974851438007179
83	0,974851271114174	0,974851509532753	1111100110001111101111001010 1	0,974851485597763
84	0,974851271114174	0,974851509532753	1111100110001111101111001010 1	0,974851485690895
85	0,974851509532753	0,974851509532753	1111100110001111101111001010 1	0,974851509532753
86	0,849851509416338	0,974851509532753	1111100110001111101111001010 1	0,962351509521111
87	0,849851509416338	0,974851509532753	1111100110001111101111001010 1	0,962351509521111
88	0,974729439220139	0,974851509532753	1111100110001111101111001010 1	0,974839302501491
89	0,974729439220139	0,974851509532753	1111100110001111101111001010 1	0,97482709547023
90	0,974729435494849	0,974851509532753	1111100110001111101111001010 1	0,974827071255843
91	0,974729439220139	0,974851509532753	1111100110001111101111001010 1	0,974839302501491
92	0,974851509532753	0,974851509532753	1111100110001111101111001010 1	0,974851509532753
93	0,974851509532753	0,974851509532753	1111100110001111101111001010 1	0,974851509532753

94	0,974851509532753	0,974851509532753	11111001100011111101111001010 1	0,974851509532753
95	0,974851390323463	0,974851509532753	11111001100011111101111001010 1	0,974851497611824
96	0,974851509532753	0,974851509532753	11111001100011111101111001010 1	0,974851509532753
97	0,974851509532753	0,974851509532753	11111001100011111101111001010 1	0,974851509532753
98	0,970945259529115	0,974851509532753	11111001100011111101111001010 1	0,974460884532389
99	0,974851509532753	0,974851509532753	11111001100011111101111001010 1	0,974851509532753
100	0,974851509532753	0,974851509532753	11111001100011111101111001010 1	0,974851509532753

## 200 ciclos sin elitismo



Algoritmo finalizado: se cumplieron los ciclos solicitados  
 Cromosoma maximo de toda la corrida: [1, 1, 1, 1, 1, 1, 0, 0, 1, 0, 1, 0, 0, 1, 1, 0, 1, 0, 1, 1, 0, 1, 1, 1]  
 Funcion objetivo del cromosoma maximo de toda la corrida: 0.9856466874346572

Ciclo	Minimo	Maximo	Cromosoma maximo	Promedio
1	0,116287680451095	0,869415006478704	11011110100100011111101101011 0	0,461083050315346
2	0,026611288102913	0,709510924955374	10110101101000101000001000001	0,253231824984059
3	0,016962391340139	0,709511252780921	10110101101000101000011110001 0	0,196476971634176

4	0,016845663093818	0,241287352741964	00111101110001010000001000001 0	0,142266210953022
5	0,016845663093818	0,241289478951403	00111101110001010010010110110 1	0,151429129905467
6	0,016845663093818	0,237548788299364	00111100110011111111111101010 1	0,16102809641606
7	0,01684556623627	0,237548691441816	00111100110011111111110110110 1	0,150871894090317
8	0,141845670660814	0,23754343878249	00111100110011111010010110010 1	0,162382620165481
9	0,141845670660814	0,155351978871368	00100111110001010010010110110 1	0,143449525854969
10	0,141845670660814	0,155351978871368	00100111110001010010010110110 1	0,145967634530708
11	0,141845663210233	0,155352075728916	00100111110001010010011101010 1	0,145967644216463
12	0,141962393319199	0,405352075961746	01100111110001010010011101010 1	0,168301418580396
13	0,141962393319199	0,141962488314102	00100100010101111010011101010 1	0,141962478814612
14	0,141962393319199	0,141962488314102	00100100010101111010011101010 1	0,141962478814612
15	0,141962393319199	0,141962488314102	00100100010101111010011101010 1	0,141962469315121
16	0,141962484588812	0,141962488314102	00100100010101111010011101010 1	0,141962487941573
17	0,141962484588812	0,141962488314102	00100100010101111010011101010 1	0,141962487941573
18	0,141962484588812	0,141962488314102	00100100010101111010011101010 1	0,141962487941573
19	0,141718347688875	0,141962488314102	00100100010101111010011101010 1	0,141937883516716
20	0,141960580965467	0,141962488314102	00100100010101111010011101010 1	0,141961916109512
21	0,141960580965467	0,149774988321378	00100110010101111010011101010 1	0,142743356845103
22	0,140984018464558	0,391962488546933	01100100010101111010011101010 1	0,16686406914784
23	0,140984018464558	0,391962488546933	01100100010101111010011101010 1	0,16686406914784
24	0,141960580965467	0,391962488546933	01100100010101111010011101010 1	0,166961725397931
25	0,141952951570929	0,391962488546933	01100100010101111010011101010 1	0,191961343951487
26	0,141952951570929	0,391962488546933	01100100010101111010011101010 1	0,19196039027717
27	0,141962488314102	0,391962488546933	01100100010101111010011101010 1	0,265790613429426
28	0,141962488314102	0,391962488546933	01100100010101111010011101010 1	0,317353113477447
29	0,141962488314102	0,391962488546933	01100100010101111010011101010 1	0,316962488477083
30	0,141962488314102	0,391962488546933	01100100010101111010011101010 1	0,2919624884538

31	0,141962488314102	0,391962488546933	01100100010101111010011101010 1	0,291864832203709
32	0,141947229525025	0,391962488546933	01100100010101111010011101010 1	0,254259546524155
33	0,141947229525025	0,391962488546933	01100100010101111010011101010 1	0,266570337551246
34	0,140970667024116	0,391962488546933	01100100010101111010011101010 1	0,316277368847539
35	0,141962488314102	0,391962488546933	01100100010101111010011101010 1	0,366156824460399
36	0,390985926046023	0,391962488546933	01100100010101111010011101010 1	0,391278870954438
37	0,390985687627444	0,391962488546933	01100100010101111010011101010 1	0,391278870954438
38	0,390985926046023	0,391962488546933	01100100010101111010011101010 1	0,391376551046387
39	0,390985926046023	0,391962488546933	01100100010101111010011101010 1	0,391278894796296
40	0,390985866441378	0,391962488546933	01100100010101111010011101010 1	0,391474201336013
41	0,390985911144862	0,391962488546933	01100100010101111010011101010 1	0,391474199845897
42	0,375360926031471	0,391962488546933	01100100010101111010011101010 1	0,389710286995126
43	0,375360926031471	0,391962488546933	01100100010101111010011101010 1	0,38990560247554
44	0,375360926031471	0,391962488546933	01100100010101111010011101010 1	0,389801842709819
45	0,375360926031471	0,391962488546933	01100100010101111010011101010 1	0,389801842709819
46	0,375360926031471	0,391962488546933	01100100010101111010011101010 1	0,389814048064699
47	0,375360926031471	0,391962488546933	01100100010101111010011101010 1	0,38805623258283
48	0,37536091113031	0,391962488546933	01100100010101111010011101010 1	0,387470298062517
49	0,37536091113031	0,391962488546933	01100100010101111010011101010 1	0,386005454311152
50	0,375360926031471	0,391962488546933	01100100010101111010011101010 1	0,386786702821764
51	0,37536091113031	0,391962488546933	01100100010101111010011101010 1	0,383850911803405
52	0,376337473631219	0,391962488546933	01100100010101111010011101010 1	0,385712482580647
53	0,141962488314102	0,391962488546933	01100100010101111010011101010 1	0,35914693228821
54	0,142450769564557	0,391962488546933	01100100010101111010011101010 1	0,359194236024464
55	0,142450769564557	0,423212488576036	01101100010101111010011101010 1	0,365448053707786
56	0,376337488532381	0,423212488576036	01101100010101111010011101010 1	0,390399988545477
57	0,376337488532381	0,423212488576036	01101100010101111010011101010 1	0,396649988551298

58	0,376337488532381	0,423212488576036	01101100010101111010011101010 1	0,402899988557119
59	0,376337488532381	0,423212488576036	01101100010101111010011101010 1	0,404462488558574
60	0,376337488532381	0,423212488576036	01101100010101111010011101010 1	0,398603114205043
61	0,376337488532381	0,423212496026617	01101100010101111010011101110 1	0,386493740777014
62	0,376337488532381	0,391962488546933	01100100010101111010011101010 1	0,382196864282896
63	0,376337488532381	0,391962488546933	01100100010101111010011101010 1	0,382196865027954
64	0,376337488532381	0,391962488546933	01100100010101111010011101010 1	0,381415615772284
65	0,376337488532381	0,391962488546933	01100100010101111010011101010 1	0,382978114283623
66	0,376337488532381	0,39586873855057	01100101010101111010011101010 1	0,384149988539657
67	0,376337488532381	0,39586873855057	01100101010101111010011101010 1	0,387665613542931
68	0,376337488532381	0,895868739016232	11100101010101111010011101010 1	0,433759363585859
69	0,376337488532381	0,39586873855057	01100101010101111010011101010 1	0,38141561353711
70	0,376337488532381	0,39586873855057	01100101010101111010011101010 1	0,382978113538565
71	0,376337488532381	0,39586873855057	01100101010101111010011101010 1	0,38141561353711
72	0,376337250113801	0,39586873855057	01100101010101111010011101010 1	0,379853092674029
73	0,376337250113801	0,391962488546933	01100100010101111010011101010 1	0,379462443831807
74	0,376337488532381	0,391962488546933	01100100010101111010011101010 1	0,381024994497211
75	0,376337488532381	0,391962488546933	01100100010101111010011101010 1	0,380243744496483
76	0,376337488532381	0,391962488546933	01100100010101111010011101010 1	0,37868124747526
77	0,376337488532381	0,391962518349255	01100100010101111010011111010 1	0,378681243004912
78	0,376337488532381	0,391962488546933	01100100010101111010011101010 1	0,379462493005639
79	0,375360926031471	0,391962488546933	01100100010101111010011101010 1	0,379364838245665
80	0,375360926031471	0,391962518349255	01100100010101111010011111010 1	0,380927339737236
81	0,375360926031471	0,391962503448094	01100100010101111010011110010 1	0,382392179018252
82	0,375360926031471	0,391962503448094	01100100010101111010011110010 1	0,380927339737236
83	0,375482996344085	0,39196344222125	01100100010101111011011101010 1	0,384064640648723
84	0,376337488532381	0,39196344222125	01100100010101111011011101010 1	0,388837588381802

85	0,376337488532381	0,39196344222125	011001000101011110110111010101 1	0,388837774646317
86	0,376337488532381	0,39196344222125	011001000101011110110111010101 1	0,388837870013749
87	0,376337488532381	0,39196344222125	011001000101011110110111010101 1	0,384162577040645
88	0,376337488532381	0,39196344222125	011001000101011110110111010101 1	0,385761888870748
89	0,376337488532381	0,392451723471705	011001000111011110110111010101 1	0,38886888873658
90	0,376337488532381	0,392451723471705	011001000111011110110111010101 1	0,388935562026627
91	0,376337488532381	0,392451723471705	011001000111011110110111010101 1	0,387373157392603
92	0,141962488314102	0,392451708570543	011001000111011110110111000101 1	0,363935655880659
93	0,376338442206698	0,392451723471705	011001000111011110110111010101 1	0,390497631384523
94	0,376338322997408	0,39196344222125	011001000101011110110111010101 1	0,388837571990525
95	0,376338322997408	0,39196344222125	011001000101011110110111010101 1	0,388837273967301
96	0,376338322997408	0,39196344222125	011001000101011110110111010101 1	0,390393777741486
97	0,376338322997408	0,395862062830349	0110010101010111100110111010101 1	0,390777727021629
98	0,391902407064943	0,395869692224888	011001010101011110110111010101 1	0,392341001883467
99	0,391902407064943	0,395808657068581	011001010101001110110111010101 1	0,392340143576581
100	0,391902407064943	0,395869692224888	011001010101011110110111010101 1	0,393878420250377
101	0,391902407064943	0,458308657126789	011101010101001110110111010101 1	0,400122310780103
102	0,391902407064943	0,458308657126789	011101010101001110110111010101 1	0,419262941758393
103	0,391955812826711	0,458308657126789	011101010101001110110111010101 1	0,425712831901119
104	0,391955812826711	0,891963442686911	111001000101011110110111010101 1	0,469658525911773
105	0,391955812826711	0,891963442686911	111001000101011110110111010101 1	0,505809267522589
106	0,391955812826711	0,891963442686911	111001000101011110110111010101 1	0,468297823395857
107	0,391955812826711	0,895808657534242	111001010101001110110111010101 1	0,475714357826593
108	0,391955812826711	0,895869692690549	111001010101011110110111010101 1	0,456188829854306
109	0,391902407064943	0,891959627989642	111001000101011101110111010101 1	0,448002108044924
110	0,391955812826711	0,891902407530604	111001000101001110110111010101 1	0,49233842696281
111	0,391902407064943	0,891959627058319	111001000101011101110111010101 0	0,491947801589917

112	0,391902407064943	0,891902406599282	11100100010100111011011101010 0	0,491941698074286
113	0,391902407064943	0,891902407530604	11100100010100111011011101010 1	0,491941698167419
114	0,14190240590079	0,891955813292373	11100100010101110011011101010 1	0,516941692137105
115	0,391898592367674	0,893908938294192	11100100110101110011011101010 1	0,642135485114656
116	0,391898592367674	0,891955813292373	11100100010101110011011101010 1	0,741933687908513
117	0,891894778136066	0,891959627989642	11100100010101110111011101010 1	0,891927203062854
118	0,891894778136066	0,891955813292373	11100100010101110011011101010 1	0,89192033660777
119	0,891894778136066	0,891955813292373	11100100010101110011011101010 1	0,891913851622412
120	0,891894778136066	0,891955813292373	11100100010101110011011101010 1	0,891932162169304
121	0,891879519346989	0,891959627989642	11100100010101110111011101010 1	0,891919573668316
122	0,891879519346989	0,891959627989642	11100100010101110111011101010 1	0,891902789000331
123	0,891898592833335	0,891898592833335	11100100010100110111011101010 1	0,891898592833335
124	0,891898592833335	0,891898592833335	11100100010100110111011101010 1	0,891898592833335
125	0,891898592833335	0,891898592833335	11100100010100110111011101010 1	0,891898592833335
126	0,891898592833335	0,891898592833335	11100100010100110111011101010 1	0,891898592833335
127	0,891898592833335	0,895804842836973	11100101010100110111011101010 1	0,892289217833699
128	0,891898592833335	0,895804842836973	11100101010100110111011101010 1	0,892289217833699
129	0,891898592833335	0,895804842836973	11100101010100110111011101010 1	0,892289217833699
130	0,891898592833335	0,895804842836973	11100101010100110111011101010 1	0,89346109283479
131	0,891898354414756	0,897757967838792	11100101110100110111011101010 1	0,894437631493842
132	0,891898354414756	0,897757967838792	11100101110100110111011101010 1	0,894632943994024
133	0,891898354414756	0,897757967838792	11100101110100110111011101010 1	0,89482825500409
134	0,891898577932174	0,897757967838792	11100101110100110111011101010 1	0,895218878514337
135	0,891898577932174	0,897757967838792	11100101110100110111011101010 1	0,894632963365534
136	0,891898577932174	0,897757967838792	11100101110100110111011101010 1	0,893851713364806
137	0,891898577932174	0,897757967838792	11100101110100110111011101010 1	0,894047025864988
138	0,891898577932174	0,897757967838792	11100101110100110111011101010 1	0,893461089854558

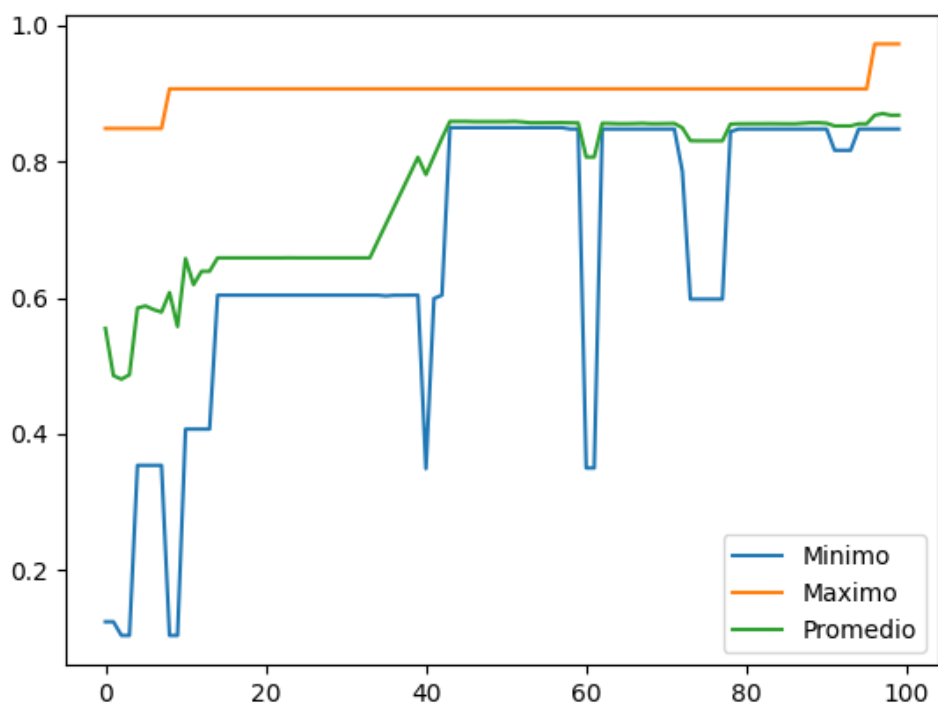


139	0,891896685484701	0,897757967838792	11100101110100110111011101010 1	0,894046826189428
140	0,891896685484701	0,897757967838792	11100101110100110111011101010 1	0,892777294938245
141	0,891898473624046	0,894828280336064	11100101000100110111011101010 1	0,893076356773336
142	0,891898473624046	0,894828280336064	11100101000100110111011101010 1	0,89249041927279
143	0,766896685368285	0,894828280336064	11100101000100110111011101010 1	0,879996296279129
144	0,766896685368285	0,923148473653149	11101100010100110111010101010 1	0,882535179967559
145	0,76689847350763	0,923148473653149	11101100010100110111010101010 1	0,882534822246558
146	0,76689847350763	0,923207720670111	11101100010101110101011101010 1	0,885720464015119
147	0,76689662576364	0,923209508809456	11101100010101110111010101010 1	0,891927912264995
148	0,76689662576364	0,923209508809456	11101100010101110111010101010 1	0,841915502624507
149	0,76689662576364	0,923209508809456	11101100010101110111010101010 1	0,876308646310408
150	0,76689662576364	0,923207720670111	11101100010101110101011101010 1	0,907564761776072
151	0,798207720553696	0,923209508809456	11101100010101110111010101010 1	0,910695871255077
152	0,798207720553696	0,923209508809456	11101100010101110111010101010 1	0,910701795956773
153	0,923146685513804	0,923209508809456	11101100010101110111010101010 1	0,923195871266719
154	0,923146685513804	0,923209508809456	11101100010101110111010101010 1	0,923195871266719
155	0,923146685513804	0,92516084567193	11101100110101110101011101010 1	0,923391183766901
156	0,923146685513804	0,92516084567193	11101100110101110101011101010 1	0,923586138639214
157	0,923146685513804	0,92516084567193	11101100110101110101011101010 1	0,923403033170293
158	0,923146685513804	0,92516084567193	11101100110101110101011101010 1	0,923378619107771
159	0,923146685513804	0,92516084567193	11101100110101110101011101010 1	0,923360308560878
160	0,923146685513804	0,92516084567193	11101100110101110101011101010 1	0,923757037076873
161	0,923146685513804	0,92516084567193	11101100110101110101011101010 1	0,924159869108498
162	0,923146685513804	0,92516084567193	11101100110101110101011101010 1	0,924367388639942
163	0,923146685513804	0,92516084567193	11101100110101110101011101010 1	0,923970660123947
164	0,923146685513804	0,92516084567193	11101100110101110101011101010 1	0,923775347623765
165	0,923146685513804	0,925099810515623	11101100110100110101011101010 1	0,923390826139032

166	0,923146685513804	0,925099810515623	111011001101001101011101010 1	0,923378619852829
167	0,923146447095225	0,925099817966204	111011001101001101011101110 1	0,923366389911032
168	0,923146447095225	0,923207728120692	11101100010101110101011101110 1	0,923169503848226
169	0,922902544888577	0,923207720670111	11101100010101110101011101010 1	0,923142060752159
170	0,891896447066121	0,92319246374368	11101100010101100101011101011 1	0,920010909549902
171	0,891896447066121	0,92319246374368	11101100010101100101011101011 1	0,916856941875869
172	0,922902544888577	0,92319246374368	11101100010101100101011101011 1	0,923111542056419
173	0,922948321255807	0,985646687434657	11111100010100110101011101011 1	0,92936609567084
174	0,922902306469998	0,985646687434657	11111100010100110101011101011 1	0,935631354652002
175	0,922902544888577	0,985646449016078	11111100010100110101001101011 1	0,92938593377302
176	0,922902544888577	0,924855669890396	11101100110000110101011101010 1	0,923253473940541
177	0,922902544888577	0,923192461881034	11101100010101100101011101010 1	0,923058137598502
178	0,922902544888577	0,923634966764259	11101100011100110101011101010 1	0,923151192928805
179	0,922902546751222	0,92314668737645	11101100010100110101011101011 1	0,92307334982145
180	0,921925984250313	0,92314668737645	11101100010100110101011101011 1	0,922926841604455
181	0,921925984250313	0,92314644895787	11101100010100110101001101011 1	0,92295123182512
182	0,423146448492209	0,92314668737645	11101100010100110101011101011 1	0,872853575528463
183	0,921925984250313	0,92314668737645	11101100010100110101011101011 1	0,922902475225648
184	0,922902546751222	0,92314668737645	11101100010100110101011101011 1	0,922987948286336
185	0,922902546751222	0,92314668737645	11101100010100110101011101011 1	0,922951374876268
186	0,922902427541932	0,92314668737645	11101100010100110101011101011 1	0,922926948892816
187	0,922902427541932	0,92314668737645	11101100010100110101011101011 1	0,922975777017862
188	0,922902427541932	0,92314668737645	11101100010100110101011101011 1	0,922926936971887
189	0,922902427541932	0,92314668737645	11101100010100110101011101011 1	0,922926936971887
190	0,922902427541932	0,927052818170798	11101101010100110101010101011 1	0,923323665487882
191	0,922902546751222	0,927052818170798	11101101010100110101010101011 1	0,923738704550768
192	0,922902427541932	0,927052937380087	11101101010100110101011101011 1	0,923738704550768

193	0,922902546751222	0,927052937380087	11101101010100110101011101011 1	0,923744819987328
194	0,922902308332643	0,930715046758498	11101110010000110101011101011 1	0,924519942630567
195	0,922902306469998	0,927052937380087	11101101010100110101011101011 1	0,924568782490277
196	0,922902306469998	0,927052937380087	11101101010100110101011101011 1	0,924153719399268
197	0,922902306469998	0,927052937380087	11101101010100110101011101011 1	0,924581037205254
198	0,922902546751222	0,927114449373553	11101101010101110101111101011 1	0,924581299651956
199	0,922902546751222	0,927114389768908	11101101010101110101111001011 1	0,92541776143519
200	0,922904452237212	0,927113972536394	11101101010101110101011101011 1	0,924587832134765

### 100 ciclos con elitismo



Algoritmo finalizado: se cumplieron los ciclos solicitados  
 Cromosoma maximo de toda la corrida: [1, 1, 1, 1, 1, 0, 0, 1, 0, 0, 0, 0, 1, 0, 1, 1, 1, 1, 1, 1, 0, 1, 0, 0, 0, 0, 0, 0]  
 Funcion objetivo del cromosoma maximo de toda la corrida: 0.9728389987469083

Ciclo	Minimo	Maximo	Cromosoma maximo	Promedio
1	0,12377076141887 4	0,848694079414656	110110010100010000000011111000	0,554871188900314
2	0,12376845080756 4	0,848694079414656	110110010100010000000011111000	0,485553352707581

3	0,10366243971852 8	0,848694079414656	110110010100010000000011111000	0,480206191148801
4	0,10366253843872 1	0,848694079414656	110110010100010000000011111000	0,486808247665696
5	0,35366243995135 9	0,848694079414656	110110010100010000000011111000	0,58481487770082
6	0,35366244833326 2	0,848694079414656	110110010100010000000011111000	0,587884584151101
7	0,35366241853093 9	0,848694079414656	110110010100010000000011111000	0,582575328631862
8	0,35366241853093 9	0,848694079414656	110110010100010000000011111000	0,578813893514512
9	0,10368736284197 1	0,906901958311817	111010000010101010111010000010	0,607577846485728
10	0,10368860708893 2	0,906901958311817	111010000010101010111010000010	0,557579279744606
11	0,40689480622009 8	0,906901958311817	111010000010101010111010000010	0,657579155413042
12	0,40689480622009 8	0,906901958311817	111010000010101010111010000010	0,61914917996074
13	0,40689480622009 8	0,906901958311817	111010000010101010111010000010	0,638829875121666
14	0,40689480622009 8	0,906901958311817	111010000010101010111010000010	0,638834557066518
15	0,60368693117395 7	0,906901958311817	111010000010101010111010000010	0,658513812775271
16	0,60368883852259 1	0,906901958311817	111010000010101010111010000010	0,658514280671733
17	0,60369308535353 6	0,906901958311817	111010000010101010111010000010	0,658514367098468
18	0,60369308535353 6	0,906901958311817	111010000010101010111010000010	0,658514219576972
19	0,60369308535353 6	0,906901958311817	111010000010101010111010000010	0,658514376039165
20	0,60369308535353 6	0,906901958311817	111010000010101010111010000010	0,658514280671733
21	0,60369308535353 6	0,906901958311817	111010000010101010111010000010	0,658514280671733
22	0,60369307045237 5	0,906901958311817	111010000010101010111010000010	0,658514332825797
23	0,60369301084773	0,906901958311817	111010000010101010111010000010	0,658514386469977
24	0,60369307045237 5	0,906901958311817	111010000010101010111010000010	0,658612476343859
25	0,60369360689417 8	0,906901958311817	111010000010101010111010000010	0,658514493758338
26	0,60369360689417 8	0,906901958311817	111010000010101010111010000010	0,658514493758338
27	0,60369360689417 8	0,906901958311817	111010000010101010111010000010	0,658514495248454
28	0,60369360689417 8	0,906901958311817	111010000010101010111010000010	0,658514878208297
29	0,60369360689417 8	0,906901958311817	111010000010101010111010000010	0,658515258187908

30	0,603693606894178	0,906901958311817	111010000010101010111010000010	0,65851526116814
31	0,603693621795339	0,906901958311817	111010000010101010111010000010	0,658514881188529
32	0,603693621795339	0,906901958311817	111010000010101010111010000010	0,65851526116814
33	0,603693621795339	0,906901958311817	111010000010101010111010000010	0,658517161066194
34	0,603693621795339	0,906901958311817	111010000010101010111010000010	0,658517149145265
35	0,603693621795339	0,906901958311817	111010000010101010111010000010	0,683517149168548
36	0,602600329185464	0,906901958311817	111010000010101010111010000010	0,708016814950869
37	0,603577258627468	0,906901958311817	111010000010101010111010000010	0,73273624548012
38	0,603577258627468	0,906901958311817	111010000010101010111010000010	0,757247929048955
39	0,603697054650352	0,906901958311817	111010000010101010111010000010	0,781771667098414
40	0,603697436492608	0,906901958311817	111010000010101010111010000010	0,806283447897326
41	0,34881462375523	0,906901958311817	111010000010101010111010000010	0,780990441498338
42	0,598814623988061	0,906901958311817	111010000010101010111010000010	0,807943605266459
43	0,603697293068932	0,906901958311817	111010000010101010111010000010	0,834414550507827
44	0,84979105261135	0,906901958311817	111010000010101010111010000010	0,859030030536493
45	0,84979105261135	0,906901958311817	111010000010101010111010000010	0,859017837475071
46	0,84979105261135	0,906901958311817	111010000010101010111010000010	0,859017809535393
47	0,84979105261135	0,906901958311817	111010000010101010111010000010	0,858627184535029
48	0,84979105261135	0,906901958311817	111010000010101010111010000010	0,858627279902461
49	0,84979105261135	0,906901958311817	111010000010101010111010000010	0,858627375269893
50	0,849791192309736	0,906901958311817	111010000010101010111010000010	0,858627389239732
51	0,84979117182064	0,906901958311817	111010000010101010111010000010	0,858627307842138
52	0,84979105261135	0,906901958311817	111010000010101010111010000010	0,859017920921573
53	0,84979105261135	0,906901958311817	111010000010101010111010000010	0,858236765915749
54	0,849791184859156	0,906901958311817	111010000010101010111010000010	0,857065174409249
55	0,849791192309736	0,906901958311817	111010000010101010111010000010	0,857065023907521
56	0,849791192309736	0,906901958311817	111010000010101010111010000010	0,857113689609909

57	0,84979119230973 6	0,906901958311817	111010000010101010111010000010	0,857162613102386
58	0,84979119230973 6	0,906901958311817	111010000010101010111010000010	0,85721134586
59	0,84783902098223 5	0,906901958311817	111010000010101010111010000010	0,856967484631545
60	0,84783902098223 5	0,906901958311817	111010000010101010111010000010	0,856772269733969
61	0,34979212316665 1	0,906901958311817	111010000010101010111010000010	0,806674801191944
62	0,34979212316665 1	0,906901958311817	111010000010101010111010000010	0,806674708059686
63	0,84783899863049 3	0,906901958311817	111010000010101010111010000010	0,856284276820984
64	0,84783899863049 3	0,906901958311817	111010000010101010111010000010	0,856088964320802
65	0,84783899863049 3	0,906901958311817	111010000010101010111010000010	0,855698339320438
66	0,84783899863049 3	0,906901958311817	111010000010101010111010000010	0,855698339320438
67	0,84783902098223 5	0,906901958311817	111010000010101010111010000010	0,85589365182062
68	0,84783899863049 3	0,906901958311817	111010000010101010111010000010	0,856284272350636
69	0,84783899863049 3	0,906901958311817	111010000010101010111010000010	0,855698337085264
70	0,84783899863049 3	0,906901958311817	111010000010101010111010000010	0,85569833485009
71	0,84783899863049 3	0,906901958311817	111010000010101010111010000010	0,855893642879924
72	0,84783899863049 3	0,906901958311817	111010000010101010111010000010	0,856088953144931
73	0,78533899857228 5	0,906901958311817	111010000010101010111010000010	0,849643638403754
74	0,59783899839766 2	0,906901958311817	111010000010101010111010000010	0,83069832588611
75	0,59783899839766 2	0,906901958311817	111010000010101010111010000010	0,830503013385928
76	0,59783900584824 3	0,906901958311817	111010000010101010111010000010	0,830503014130986
77	0,59783899839766 2	0,906901958311817	111010000010101010111010000010	0,830503014130986
78	0,59783899839766 2	0,906901958311817	111010000010101010111010000010	0,830503014130986
79	0,84393274862685 5	0,906901958311817	111010000010101010111010000010	0,855112388408848
80	0,84783899863049 3	0,906901958311817	111010000010101010111010000010	0,855551841534257
81	0,84783899863049 3	0,906901958311817	111010000010101010111010000010	0,855600669659302
82	0,84783899863049 3	0,906901958311817	111010000010101010111010000010	0,855649497784348
83	0,84783899863049 3	0,906901958311817	111010000010101010111010000010	0,855649497784348

84	0,84783899863049 3	0,906901958311817	111010000010101010111010000010	0,855747154034439
85	0,84783899863049 3	0,906901958311817	111010000010101010111010000010	0,855655601299979
86	0,84783899863049 3	0,906901958311817	111010000010101010111010000010	0,855570152081149
87	0,84783518393322 4	0,906901958311817	111010000010101010111010000010	0,855520942486377
88	0,84783518393322 4	0,906901958311817	111010000010101010111010000010	0,856296088971473
89	0,84783518579586 9	0,906901958311817	111010000010101010111010000010	0,857077339158466
90	0,84783518579586 9	0,906901958311817	111010000010101010111010000010	0,857077339158466
91	0,84783518579586 9	0,906901958311817	111010000010101010111010000010	0,856289985642107
92	0,81658899860138 9	0,906901958311817	111010000010101010111010000010	0,852378013406301
93	0,81658899860138 9	0,906901958311817	111010000010101010111010000010	0,852378013406301
94	0,81658899860138 9	0,906901958311817	111010000010101010111010000010	0,852378013406301
95	0,84783518393322 4	0,906901958311817	111010000010101010111010000010	0,855502631939484
96	0,84783518393322 4	0,906901958311817	111010000010101010111010000010	0,855502250469758
97	0,84783899863049 3	0,972838998746908	111110010000101111111010000000	0,868100669670944
98	0,84783899863049 3	0,972838998746908	111110010000101111111010000000	0,870639638482258
99	0,84783899863049 3	0,972838998746908	111110010000101111111010000000	0,868100669857208
100	0,84783899863049 3	0,972838998746908	111110010000101111111010000000	0,868198326293564

## Conclusiones

- Al aumentar el número de ciclos, aumenta la probabilidad de encontrar una población con cromosomas de un valor mayor.
- Se observa que sin elitismo, en algunas corridas el valor del máximo cromosoma de la última población es inferior a algunos máximos obtenidos previamente
- Al activar elitismo, con el paso de los ciclos, los valores aumentan en menor

proporción ya que los cromosomas que forman parte de la élite van mejorando cada vez más y se hace muy difícil que surja un cromosoma nuevo con un mayor valor a los de la misma.

- Al activar elitismo, los valores máximos de la función nunca disminuyen y tienden a 1 cuando finaliza la ejecución
- Aumentar los valores de mutación incrementa mucho las variaciones de los valores y permite que el algoritmo no converja tan rápidamente.
- Si los valores de mutación se aumentan demasiado, aumenta la probabilidad de que el algoritmo pierda cromosomas con altos valores que podrían ser utilizados para mejorar los resultados obtenidos en las posteriores poblaciones, haciendo que al final de la ejecución del programa no se logre obtener el cromosoma óptimo en la población.
- Al disminuir la cantidad de cromosomas en la población, la probabilidad de obtener al cromosoma óptimo en la población es casi nula.
- Al aumentar la cantidad de cromosomas con elitismo, los valores tienden a variar menos hasta que, al llegar al tamaño de la población, los valores de la función dejan de evolucionar.
- Al disminuir la probabilidad de crossover, los cromosomas varían menos hasta que, cuando la probabilidad llega al 0%, los únicos cambios que se observan en los cromosomas son los realizados por las mutaciones.
- Al realizar pocos ciclos las probabilidades de encontrar al óptimo en la población son muy bajas.