

Métodos de Inteligencia Artificial

Reporte 5

IF698972

Josefina Esmeralda Arriaga Hernández

2 de febrero del 2017 Guadalajara, Jalisco

**Objetivo**

Crear un programa de un algoritmo genético con el propósito de aproximarse a las funciones establecidas.

**Problema a resolver**

Se tiene las funciones f(x, y) = -20 exp(-0.2 (.5 (x2 + y2) )1/2) - exp (0.5 (cos (2πx) + cos (2πy) ) ) +e+20 dentro del intervalo -5 ≤ x ≤ 5 que se va a minimizar y f(x, y) = 2x2 - 1.05 x4 + x6 /6 + xy + y2 dentro del intervalo -5 ≤ x ≤ 5 se minimizará, se va a desarrollar un algoritmo genético a partir de estos.

Los pasos para crear el algoritmo son:

1. Generar la población.
2. Se evalúa los pobladores.
3. Se selecciona los progenitores.
4. Se realiza el cruzamiento para obtener hijos.
5. Se muta a hijos aleatorios.
6. Se crea una iteración para llegar al resultado deseado.
7. Por último, se promedia el eje y y se busca el mínimo.

**Código desarrollado**

clear all;

close all;

clc

np = 32; %N?mero de pobladores

%Para primera x

tpaso = .01; %Tama?o de paso

xmin = -5;

xmax = 5;

elementos = (xmax - xmin)/ tpaso; %K

nbits = ceil(log2(elementos));

%Para segunda x

tpaso2 = .01; %Tama?o de paso

xmin2 = -5;

xmax2 = 5;

elementos2 = (xmax2 - xmin2)/ tpaso2; %K

nbits2 = ceil(log2(elementos2));

%Generamos la poblaci?n inicial

x1 = randi([0,2^nbits-1],np,1); %Genera np n?meros aleatorios enteros en una columna

x1real = ((xmax - xmin)/(2^nbits-1))\*x1 + xmin; %Convierte enteros a decimales

%Generamos la poblaci?n inicial

x2 = randi([0,2^nbits2-1],np,1); %Genera np n?meros aleatorios enteros en una columna

x2real = ((xmax2 - xmin2)/(2^nbits2-1))\*x2 + xmin2; %Convierte enteros a decimales

%(funci?n objetivo)

for i = 1:100 %N?mero de generaciones

    y = -(-20\*exp(-.2\*sqrt(.5\*(x1real+x2real).^2))-exp(.5\*(cos(2\*pi.\*x1real)+cos(2\*pi.\*x2real)))+euler(1)+20);

    %y = - (2.\*(x1real).^2-1.05.\*(x1real).^4+(x1real.^6)/6+(x1real.\*x2real)+x2real.^2);

    yprom(i) = mean(y); %Media del vector y

    cromosoma = [y x1 x1real x2 x2real];

    cromosomaOrd = sortrows(cromosoma,1); %Ordena en funci?n de la columna 1

    %Selecci?n

    padres = cromosomaOrd(np/2+1:np,2);

    padresbin = dec2bin(padres,nbits);

    padres2 = cromosomaOrd(np/2+1:np,4);

    padresbin2 = dec2bin(padres2,nbits2);

    %Cruzamiento

    %Un punto de cruce p

    for k = 1:(np/4)

        %Hijo1

        p = randi([2 nbits-1]); %Aleatorios enteros del 2 a nbits-1

        hijobin(2\*k-1,:) = [padresbin(2\*k-1,1:p) padresbin(2\*k,p+1:nbits)];

        %Hijo2

        p = randi([2 nbits-1]);

        hijobin(2\*k,:) = [padresbin(2\*k,1:p) padresbin(2\*k-1,p+1:nbits)];

    end

     for k = 1:(np/4)

        %Hijo1

        p = randi([2 nbits2-1]); %Aleatorios enteros del 2 a nbits-1

        hijobin2(2\*k-1,:) = [padresbin2(2\*k-1,1:p) padresbin2(2\*k,p+1:nbits2)];

        %Hijo2

        p = randi([2 nbits-1]);

        hijobin2(2\*k,:) = [padresbin2(2\*k,1:p) padresbin2(2\*k-1,p+1:nbits2)];

    end

    %Mutaci?n

    m = rand();

    if m >= 0.8

        nhijo = randi(np/4);

        bit = randi([1,nbits]);

        if hijobin(nhijo,bit) == '1'

            hijobin(nhijo,bit) = '0';

        else

            hijobin(nhijo,bit) = '1';

        end

    end

      m = rand();

    if m >= 0.8

        nhijo = randi(np/4);

        bit = randi([1,nbits2]);

        if hijobin2(nhijo,bit) == '1'

            hijobin2(nhijo,bit) = '0';

        else

            hijobin2(nhijo,bit) = '1';

        end

    end

    %Convertir a n?meros reales en base a la funci?n objetivo

    hijodec = bin2dec(hijobin);

    hijoreal = ((xmax - xmin)/(2^nbits-1))\*hijodec + xmin;

    %Convertir a n?meros reales en base a la funci?n objetivo

    hijodec2 = bin2dec(hijobin2);

    hijoreal2 = ((xmax2 - xmin2)/(2^nbits2-1))\*hijodec2 + xmin2;

    %Juntamos padres e hijos en el vector de x1

    %Aqu? se eliminan los m?s d?biles

    x1real = [cromosomaOrd(np/2+1:np,3);hijoreal];

    x1 = [cromosomaOrd(np/2+1:np,2);hijodec];

    x2real = [cromosomaOrd(np/2+1:np,5);hijoreal2];

    x2 = [cromosomaOrd(np/2+1:np,4);hijodec2];

end

plot(yprom)

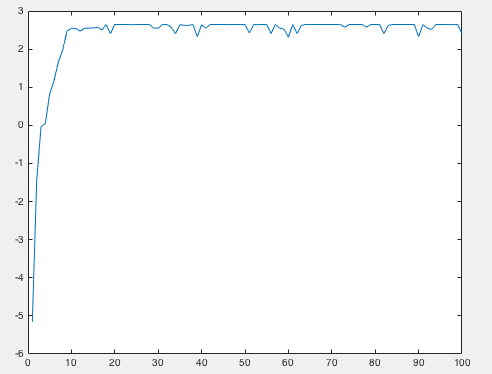
[val,ind] = max(y)

disp(['Resultado: x1= ' num2str(cromosoma(ind,3))...

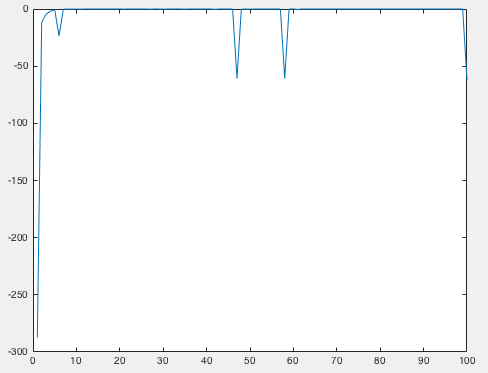
    ' x2=' num2str(cromosoma(ind,5)) ', Desempe?o = ' num2str(val)])

**Gráficos**

**Primera función**

****

**Segunda función**

****

**Interpretación de gráficos**

En las primeras iteraciones se llega al resultado deseado y con ayuda de las mutaciones que se realizan de manera aleatoria se observa que se llega el valor esperado de las x1 y x2, entonces el promedio del eje x1 y x2 da como resultado el mínimo de la función. Si no se tuviera la mutación podría salir súper individuos y los hijos saldrían igual haciendo que se estanque el promedio. En ambos casos las x esperadas se aproximan a cero.

**Resultados**

El resultado de la primera función es f(0, 0)=0, siendo este el promedio de ambas x de la función f(x, y) = -20 exp(-0.2 (.5 (x2 + y2) )1/2) - exp (0.5 (cos (2πx) + cos (2πy) ) ) +e+20, se logra obtener ese resultado ya que se agregó la mutación (cambiando el número binario de un poblador al azar) haciendo que el algoritmo no tenga súper individuos. Son 32 pobladores que van generando generaciones 1000 veces.

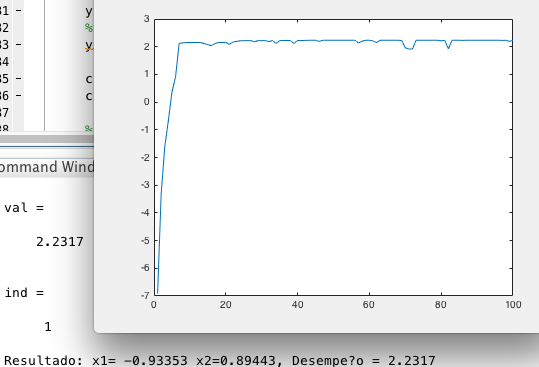
El de la segunda función es f(x, y) = 2x2 - 1.05 x4 + x6 /6 + xy + y2 en un intervalo de -5 ≤ x ≤ 5 teniendo como resultado 0 ya que éste es su mínimo de las dos x. Son 32 pobladores que van generando generaciones 1000 veces. Todos los algoritmos genéticos se realizaron con el cruzamiento de un punto. En el algoritmo genético se dice también en cual iteración se llegó al resultado esperado al igual que los resultados de ambas incógnitas.

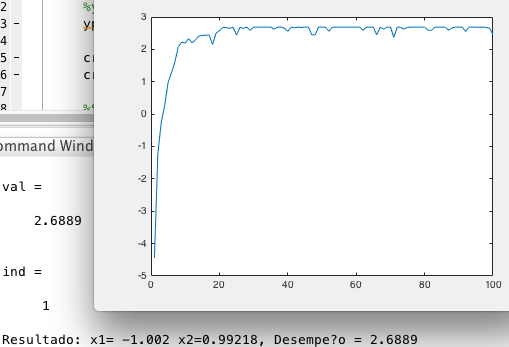
**Conclusiones**

En conclusión, se puede decir que el programa es exitoso ya que se llega al resultado esperado sin crear súper individuos, en pocas iteraciones. El algoritmo genético es heurístico siendo esto que se obtuvo una buena solución, para las distintas funciones se obtuvo el resultado deseado y se aprendió a crear algoritmos genéticos con dos variables.

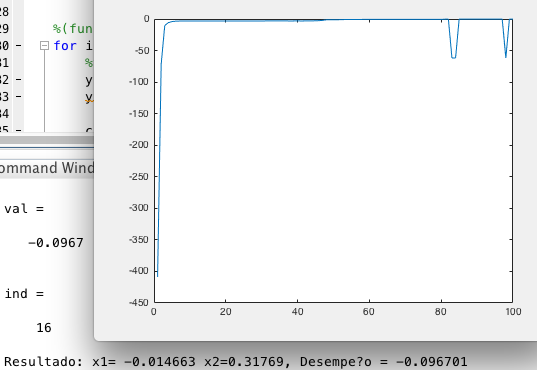
**Funcionamiento**

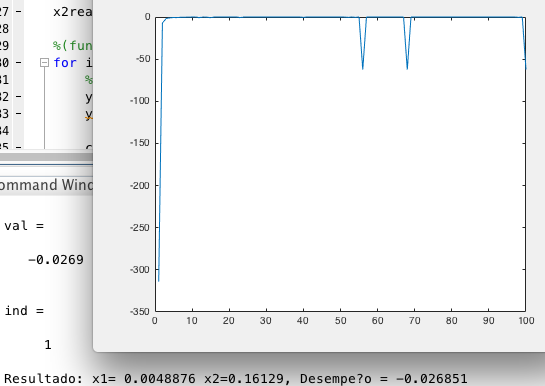
**Primera función**





**Segunda función**

****

****