**Homework** 1:

Find the minimized value of the following function:



**Solution I** ():

|  |  |
| --- | --- |
|  |  |
|  |  |

**Code** in Matlab:

% plot y=f(x)

x=0:0.01:1;

y=sin(2\*pi()\*x)+0.5\*sin(6\*pi()\*x)+0.5\*cos(10\*pi()\*x);

plot(x,y,'k','LineWidth',1.5)

xlabel('x')

ylabel('y')

title('sin(2\pix)+^{1}/\_{2}sin(6\pix)+^{1}/\_{2}cos(10\pix)')

grid on

clear x y

%solve

learning\_rate\_Array = [0.001, 0.01, 0.1, 1];

x0\_Array = [0.1, 0.4, 0.9]

iterations = 1000

for i\_x0 = 1:numel(x0\_Array)

for i\_alpha = 1:numel(learning\_rate\_Array)

learning\_rate = learning\_rate\_Array(i\_alpha);

x(1)=x0\_Array(i\_x0);

epsilon = 0.0001;

for i=1:iterations

dy(i)=-5\*pi()\*sin(10\*pi()\*x(i))+3\*pi()\*cos(6\*pi()\*x(i))+2\*pi()\*cos(2\*pi()\*x(i));

if abs(dy(i)) <= epsilon

y(i)=sin(2\*pi()\*x(i))+0.5\*sin(6\*pi()\*x(i))+0.5\*cos(10\*pi()\*x(i));

[n,xx,yy,dyy] = deal(i,x(i),y(i),dy(i));

disp(['Results for x0 = ', num2str(x(1)), ', alpha = ', num2str(learning\_rate) ...

': n = ', num2str(n), ', ', 'xx = ', num2str(xx), ...

', ', 'yy = ', num2str(yy), ', ', 'dyy = ', num2str(dyy)])

break

else

x(i+1)=x(i)-learning\_rate\*dy(i);

y(i)=sin(2\*pi()\*x(i))+0.5\*sin(6\*pi()\*x(i))+0.5\*cos(10\*pi()\*x(i));

end

if x(i+1) < 0 || x(i+1) > 1 %range of function

y(i)=sin(2\*pi()\*x(i))+0.5\*sin(6\*pi()\*x(i))+0.5\*cos(10\*pi()\*x(i));

[n,xx,yy,dyy] = deal(i,x(i),y(i),dy(i));

Disp(…)

break

end

if i == iterations

disp(…)

end

end

end

end

**Output**:

Results for x0 = 0.1, alpha = 0.001: n = 31, xx = 0.092627, yy = 0.55539, dyy = 8.8748e-05

Results for x0 = 0.1, alpha = 0.01: no results found in 1000 iterations

Results for x0 = 0.1, alpha = 0.1: n = 1, xx = 0.1, yy = 0.56331, dyy = 2.1708

Results for x0 = 0.1, alpha = 1: n = 1, xx = 0.1, yy = 0.56331, dyy = 2.1708

Results for x0 = 0.4, alpha = 0.001: n = 34, xx = 0.5338, yy = -0.75184, dyy = -9.5392e-05

Results for x0 = 0.4, alpha = 0.01: no results found in 1000 iterations

Results for x0 = 0.4, alpha = 0.1: n = 2, xx = 0.61708, yy = -0.64353, dyy = -7.089

Results for x0 = 0.4, alpha = 1: n = 1, xx = 0.4, yy = 1.5633, dyy = -2.1708

Results for x0 = 0.9, alpha = 0.001: n = 10, **xx = 0.89682, yy = -1.5668**, dyy = 7.4361e-05

Results for x0 = 0.9, alpha = 0.01: no results found in 1000 iterations

Results for x0 = 0.9, alpha = 0.1: n = 3, xx = 0.84432, yy = -0.84361, dyy = -21.1726

Results for x0 = 0.9, alpha = 1: n = 1, **xx = 0.9, yy = -1.5633**, dyy = 2.1708

When instead of gradient descent converged only in case of learning rate value depending on initial point in 31, 34 and 10 iterations pointing to local minima , , . **Other than that gradient descend did not converge**.

* Gradient descent method does not guarantee global minimum of function**,** only **local minimum** in case of convergence.

**Solution II** ():

Code is roughly the same as in the slides, only fitness function is slightly different:

function PI = GA\_fitfun(chro)

global MIN\_offset

MIN\_offset = 10;

x = chro(1);

y = sin(2\*pi()\*x)+0.5\*sin(6\*pi()\*x)+0.5\*cos(10\*pi()\*x);

PI = MIN\_offset - y;

Because GA doesn’t generate new population by crossover, GA is initialized with a rich set of solutions (initial population is 1000)

**Result**:

Real value of chromosome with the highest fitness value is **0.8980** which is indeed close to **global minimum** of function ()