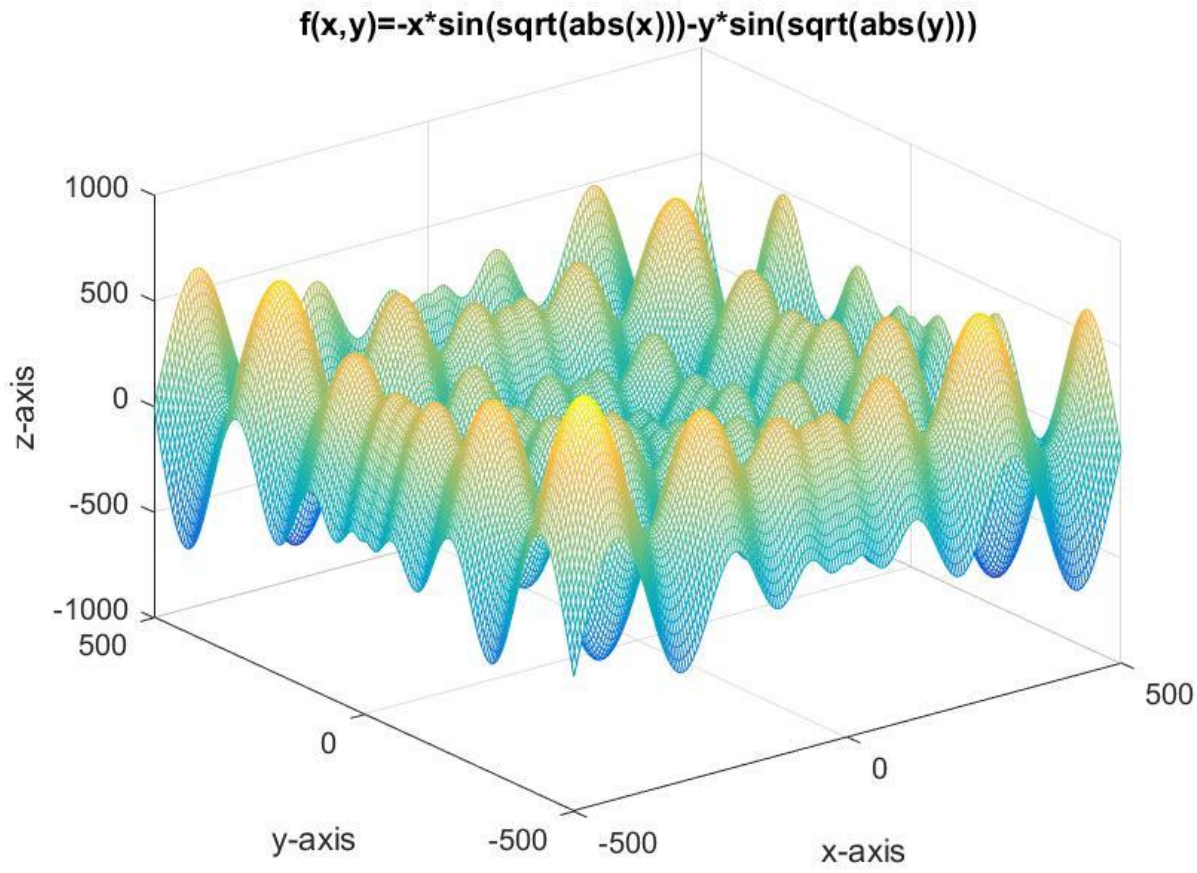
Andrew Camps

Homework 8

11/30/16

Problem 1: Function plotFun that graphs the function below



Code:

```
function plotFun( x0, x1, y0, y1 )
%plotFun Function that graphs the f(x,y) = -x.* sin(sqrt(abs(x)))
%- y.* sin(sqrt(abs(y))).


x = x0:5:x1;
y = y0:5:y1;

[x, y] = meshgrid(x, y);

fxy = -x.* sin(sqrt(abs(x))) - y.* sin(sqrt(abs(y)));

mesh(x, y, fxy);
xlabel('x-axis') %Axis and title labeling
ylabel('y-axis')
zlabel('z-axis')
title('f(x,y)=-x*sin(sqrt(abs(x)))-y*sin(sqrt(abs(y)))')

end
```

Problem 2: Simulated Annealing Code

```
function [ P_best, E_best, iteration] = SA( f, dim, domain, start, T0,
Lambda, minTemp )
%SA Function that uses the Simulated Annealing algorithm to calculate the
%global min of a functon specified.

finished = 0;
iteration = 0;
T = T0; %Initial Temp

for i=1:1:dim
    P(i) = {start(i)};
end

E_current = f(P{:}); %Calculate initial current fitness
E_best = E_current;
P_best = P;

while ~finished % Loop until exit criteria met
    iteration = iteration + 1; %Iteration counter
    PTemp = newSol(P, domain(1), domain(2)); % Calls function to modify
parameters

    for i=1:1:dim
        PNew(i) = {PTemp(i)};
    end

    E_temp = f(PNew{:}); %Gets temp

    if(E_temp < E_current) %Compares temp to current
```

```
        E_current = E_temp; %Current becomes temp
        P = PNew;
        if(E_current < E_best) %Checks for best
            E_best = E_current;
            P_best = P;
            if(E_best < minTemp)
                finished = 1;   %Exit if reached goal
            end
        end
    else
        X = rand;
        if(X < exp(-(E_temp - E_best) / T)) %Accept with a certain
probability
            E_current = E_temp;
            P = PNew;
        end
    end

    T = Lambda * T; %Cooling

    if(T < minTemp)
        finished = 1;   % Exit criteria
    end

end

end


function mu = newSol(x, Lb, Ub)
    mu = Lb + (Ub - Lb) .* rand(size(x)); % modifies parameters
end
```

Problem 3: Table with Matlab Script with different T0 and Lambda Values

	T0_300	T0_200	T0_100	T0_50	T0_10	T0_Neg20	T0_Neg100	T0_Neg150	T0_Neg200	T0_Neg250
0.1	-494.7	-627.91	-302.13	-422.64	-312.42	-320.74	-371.6	-428.73	-646.32	-671.61
0.2	-463.94	-445.89	-312.6	-318.88	-325.89	-414.51	-603.59	-528.82	-521.07	-473.88
0.3	-307.56	-450.69	-347.56	-420.77	-714.8	-347.39	-776.65	-366.94	-327.82	-339.34
0.4	-437.87	-465.68	-528.12	-411.93	-369.02	-589.53	-388.63	-559.02	-341.09	-388.05
0.5	-430.66	-454.82	-505.32	-355.21	-301.82	-366.95	-427.84	-354.75	-300.49	-333.62
0.6	-309.4	-576.48	-377.92	-311.08	-481.82	-402.01	-553.64	-784.61	-441.97	-367.5
0.7	-517.11	-309.3	-336.06	-321.31	-442.56	-316.15	-503.75	-439.34	-513.36	-641.61
0.8	-331.55	-453.19	-594.73	-373.25	-493.16	-517.58	-338.86	-314.23	-320.38	-423.15
0.9	-402.59	-310.76	-324.48	-349.53	-309.13	-736.23	-325.65	-410.83	-328.63	-327.02
1	-369.87	-361.67	-491.83	-484.89	-448.45	-600.94	-501.33	-502.42	-440.78	-642.83

Code:

```
clear

f = @(x,y)-x*sin(sqrt(abs(x))) - y*sin(sqrt(abs(y)));
dim = 2;
domain = [-500, 500];
minTemp = -300;
start = [150; -40];

Lambda = [0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1];
T0 = [300, 200, 100, 50, 10, -20, -100, -150, -200, -250];



for i = 1:1:10
    for j = 1:1:10
        [P_best(i,:,j), E_best(i,:,j)] = SA(f, dim, domain, start, T0(i),
Lambda(j), minTemp);
    end
end

RowNames = {'0.1'; '0.2'; '0.3'; '0.4'; '0.5'; '0.6'; '0.7'; '0.8'; '0.9';
'1'};
T0_300 = E_best(1,:)';
T0_200 = E_best(2,:)';
T0_100 = E_best(3,:)';
T0_50 = E_best(4,:)';
T0_10 = E_best(5,:)';
T0_Neg20 = E_best(6,:)';
T0_Neg100 = E_best(7,:)';
T0_Neg150 = E_best(8,:)';
T0_Neg200 = E_best(9,:)';
T0_Neg250 = E_best(10,:)';

table(T0_300, T0_200, T0_100, T0_50, T0_10, T0_Neg20,...
    T0_Neg100, T0_Neg150, T0_Neg200, T0_Neg250, 'RowNames', RowNames)
```

Problem 4: 10 Dimension optimization problem

Best Run Values:

Lambda = 0.9;

T0 = -2.17;

minTemp = -6.7;

start = `[1.9,1.5,2,3,2,1.4,2.6,1.6,0.3,1.6];`

E_best = -6.75547911857615

P_best = 2.09237710457881 0.739500413257148 2.07598388729929 2.06357664453415

2.00059790733111 1.53265892264426 0.578724877890683 2.42048284390297

1.43213906876917 1.70578049866120

Iteration = 8998220

Code:

```
clear
pi = 3.14159;

f = @(x1,x2,x3,x4,x5,x6,x7,x8,x9,x10) -(sin(x10)*(sin(x1^(2/pi))^(20)) +
sin(x9)*(sin(2*x2^(2/pi))^(20)) + ...
    sin(x8)*(sin(3*x3^(2/pi))^(20)) + sin(x7)*(sin(4*x4^(2/pi))^(20)) +
sin(x6)*(sin(5*x5^(2/pi))^(20)) + ...
    sin(x5)*(sin(6*x6^(2/pi))^(20)) + sin(x4)*(sin(7*x7^(2/pi))^(20)) +
sin(x3)*(sin(8*x8^(2/pi))^(20)) + ...
    sin(x2)*(sin(9*x9^(2/pi))^(20)) + sin(x1)*(sin(10*x10^(2/pi))^(20)));

dim = 10;
domain = [0,pi];
Lambda = .9;
T0 = -2.17;
minTemp = -6.7;
start = [1.9,1.5,2,3,2,1.4,2.6,1.6,0.3,1.6];

[P_best, E_best, iteration] = SA(f, dim, domain, start, T0, Lambda, minTemp);
```