ECE 580 Fun Work #4

Vineeth Ravi

March 27, 2018

PU ID : 0030019456
EMAIL ID : ravi24@purdue.edu

Answer 1

y(t) = $sin(\omega t + \theta)$

We wish to determine ω and θ , such that :-

$$sin(\omega t_i + \theta) = y_i \text{ , i =1,2...,p}$$

(a)
System of Linear Equations

$$\begin{gathered} sin(\omega t_1 + \theta) = y_1 \\ . \\ . \\ . \\ sin(\omega t_p + \theta) = y_p \end{gathered}$$

Taking sin^{-1} or $arcsin$ both sides,
we get the following system of linear equations in ω and θ

$$\begin{gathered} \omega t_1 + \theta = sin^{-1}(y_1) \\ . \\ . \\ . \\ \omega t_p + \theta = sin^{-1}(y_p) \end{gathered}$$

i.e

$$\omega t_i + \theta = sin^{-1}(y_i) \text{ , i =1,2...,p}$$

The above system of Linear Equations can be written in the form of Ax=b.

(b)

The Linear Equations got, can be written as Ax = b, where

$$A = \begin{bmatrix} t_1 & 1 \\ . & . \\ . & . \\ . & . \\ t_p & 1 \end{bmatrix} , x = \begin{bmatrix} \omega \\ \theta \end{bmatrix} , b = \begin{bmatrix} sin^{-1}(y_1) \\ . \\ . \\ . \\ sin^{-1}(y_p) \end{bmatrix}$$

Since, the t_i are not equal, the first column of A is not a scalar multiple of the second column of A.
$\therefore$ rank A = 2.

The Least Squares Solutions is given by :

$$x = (A^TA)^{-1}A^Tb$$

$$= \begin{bmatrix} \sum_{i=1}^{p} {t_i}^2 & \sum_{i=1}^{p} t_i \\ \sum_{i=1}^{p} t_i & p \end{bmatrix}^{-1} \begin{bmatrix} \sum_{i=1}^{p} t_i sin^{-1}(y_i) \\ \sum_{i=1}^{p} sin^{-1}(y_i) \end{bmatrix}$$

$$= \begin{bmatrix} \sum_{i=1}^{p} \frac{{t_i}^2}{p} & \sum_{i=1}^{p} \frac{t_i}{p} \\ \\ \sum_{i=1}^{p} \frac{t_i}{p} & 1 \end{bmatrix}^{-1} \begin{bmatrix} \sum_{i=1}^{p} \frac{t_i sin^{-1}(y_i)}{p} \\ \\ \sum_{i=1}^{p} \frac{sin^{-1}(y_i)}{p} \end{bmatrix}$$

$$= \begin{bmatrix} \overline{T^2} & \overline{T} \\ \overline{T} & 1 \end{bmatrix}^{-1} \begin{bmatrix} \overline{TY} \\ \overline{Y} \end{bmatrix}$$

$$= \frac{1}{\overline{T^2} - (\overline{T})^2} \begin{bmatrix} 1 & -\overline{T} \\ -\overline{T} & \overline{T^2} \end{bmatrix} \begin{bmatrix} \overline{TY} \\ \overline{Y} \end{bmatrix}$$

$$= \frac{1}{\overline{T^2} - (\overline{T})^2} \begin{bmatrix} \overline{TY} - (\overline{T})(\overline{Y}) \\ -(\overline{T})(\overline{TY}) + (\overline{T^2})(\overline{Y}) \end{bmatrix}$$

Hence

$$x = \begin{bmatrix} \omega \\ \theta \end{bmatrix} = \frac{1}{\overline{T^2} - (\overline{T})^2} \begin{bmatrix} \overline{TY} - (\overline{T})(\overline{Y}) \\ -(\overline{T})(\overline{TY}) + (\overline{T^2})(\overline{Y}) \end{bmatrix}$$

Q.E.D

Answer 2

We are given the discrete time system :

$$x_{k+1} = ax_k + bu_k$$
and
$$s_0 = x_0 = 0 \ , \ u_k = 1$$
$$s_1 = as_0 + b$$
$$s_2 = as_1 + b$$
$$.$$
$$.$$
$$.$$
$$s_n = as_{n-1} + b$$

To find the Least Square Estimate of $x = [a, b]^T$, we re-write the problem as :-

$$Ax = B, \text{ where}$$
$$A = \begin{bmatrix} 0 & 1 \\ s_1 & 1 \\ s_2 & 1 \\ . & . \\ . & . \\ . & . \\ s_{n-1} & 1 \end{bmatrix} , \ x = \begin{bmatrix} a \\ b \end{bmatrix} , \ B = \begin{bmatrix} s_1 \\ s_2 \\ s_3 \\ . \\ . \\ . \\ s_n \end{bmatrix}$$

The Least Squares Solutions is given by :

$$x = (A^TA)^{-1}A^TB$$

The matrix A^TA is non singular because, it is given that at least one s_k is non zero.

$$A^TA = \begin{bmatrix} \sum_{i=1}^{n-1} s_i^2 & \sum_{i=1}^{n-1} s_i \\ \\ \sum_{i=1}^{n-1} s_i & n \end{bmatrix}$$

and

$$A^TB = \begin{bmatrix} \sum_{i=1}^{n-1} s_i s_{i+1} \\ \\ \sum_{i=1}^{n} s_i \end{bmatrix}$$

$$x = (A^TA)^{-1}A^TB$$

$$= \begin{bmatrix} \sum_{i=1}^{n-1} s_i^2 & \sum_{i=1}^{n-1} s_i \\ \\ \sum_{i=1}^{n-1} s_i & n \end{bmatrix}^{-1} \begin{bmatrix} \sum_{i=1}^{n-1} s_i s_{i+1} \\ \\ \sum_{i=1}^{n} s_i \end{bmatrix}$$

$$= \frac{1}{n\sum_{i=1}^{n-1} s_i^2 - (\sum_{i=1}^{n-1} s_i)^2} \begin{bmatrix} n\sum_{i=1}^{n-1} s_i s_{i+1} - \sum_{i=1}^{n-1} s_i \sum_{i=1}^{n} s_i \\ \\ -\sum_{i=1}^{n-1} s_i \sum_{i=1}^{n-1} s_i s_{i+1} + \sum_{i=1}^{n-1} s_i^2 \sum_{i=1}^{n} s_i \end{bmatrix}$$

Hence

$$x = \begin{bmatrix} a \\ b \end{bmatrix} = \frac{1}{n\sum_{i=1}^{n-1} s_i^2 - (\sum_{i=1}^{n-1} s_i)^2} \begin{bmatrix} n\sum_{i=1}^{n-1} s_i s_{i+1} - \sum_{i=1}^{n-1} s_i \sum_{i=1}^{n} s_i \\ \\ -\sum_{i=1}^{n-1} s_i \sum_{i=1}^{n-1} s_i s_{i+1} + \sum_{i=1}^{n-1} s_i^2 \sum_{i=1}^{n} s_i \end{bmatrix}$$

Q.E.D

Answer 3

The problem given to us is :

$$\text{minimize } ||x - x_0||$$
$$\text{subject to } [1\ 1\ 1]x = 1$$

We obtain the solution to this problem, by converting it to a problem which has the Least Norm form.

General solution for the problem:-

$$\text{minimize } ||x - x_0||$$
$$\text{subject to } Ax = b \text{ , where } A \in R^{m\times n}\text{, and } rankA = m \leqslant n$$

Let $z = x - x_0$, We transform the coordinates to get the below :-

$$\text{minimize } ||z||$$
$$\text{subject to } Az = b - Ax_0$$

From Linear Algebra's least norm solution, we get that the solution is of the form :-

$$z^* = A^T(AA^T)^{-1}(b - Ax_0)$$

$$z^* = A^T(AA^T)^{-1}b - A^T(AA^T)^{-1}Ax_0$$

Transforming the coordinates back we get $x = z + x_0$

$$x^* = A^T(AA^T)^{-1}(b - Ax_0) + x_0$$

$$x^* = A^T(AA^T)^{-1}b - A^T(AA^T)^{-1}Ax_0) + x_0$$

$$x^* = A^T(AA^T)^{-1}b + (I_n - A^T(AA^T)^{-1}A)x_0$$

Hence we have obtained the general formula for the solution x^* .

Substituting A=[1 1 1], b = 1 and $x_0 = [0, -3, 0]^T$ we get :-

$$x^* = \begin{bmatrix}1\\1\\1\end{bmatrix} (\begin{bmatrix}1 & 1 & 1\end{bmatrix} \begin{bmatrix}1\\1\\1\end{bmatrix})^{-1}(1) + (\begin{bmatrix}1 & 0 & 0\\0 & 1 & 0\\0 & 0 & 1\end{bmatrix} - \begin{bmatrix}1\\1\\1\end{bmatrix} (\begin{bmatrix}1 & 1 & 1\end{bmatrix} \begin{bmatrix}1\\1\\1\end{bmatrix})^{-1} \begin{bmatrix}1 & 1 & 1\end{bmatrix}) \begin{bmatrix}0\\-3\\0\end{bmatrix}$$

$$x^* = \begin{bmatrix}\frac{1}{3}\\ \frac{1}{3}\\ \frac{1}{3}\end{bmatrix} + \begin{bmatrix}\frac{2}{3} & \frac{-1}{3} & \frac{-1}{3}\\ \frac{-1}{3} & \frac{2}{3} & \frac{-1}{3}\\ \frac{-1}{3} & \frac{-1}{3} & \frac{2}{3}\end{bmatrix} \begin{bmatrix}0\\-3\\0\end{bmatrix}$$

$$x^* = \begin{bmatrix}\frac{4}{3}\\ \frac{-5}{3}\\ \frac{4}{3}\end{bmatrix} = \begin{bmatrix}1.3333\\-1.6667\\1.3333\end{bmatrix}$$

Hence the solution is found.

$$\text{The minimum value of } ||x - x_0|| \text{ is } = ||A^T(AA^T)^{-1}b - A^T(AA^T)^{-1}Ax_0||$$

The minimum value of $||x - x_0||$ is **2.3094** .

```
% MATLAB CODE to compute the expression of x*

A = [1 1 1];
b=1;
x0=[0;3;0];


x=A'*inv(A*A')*b - (eye(3)-A'*inv(A*A')*A)*x0;

norm(x-x0)

% The END
```

Answer 4

The MATLAB Code is include after the Answer.
The Griewank function is Minimized by the PSO algorithm.
The Dimension is 2. The number of Particles in the Swarm is set to 100.

The value of the coordinates for each particle is clipped between [-5,5].

The value of the velocities for each particle is clipped between [-1,1].

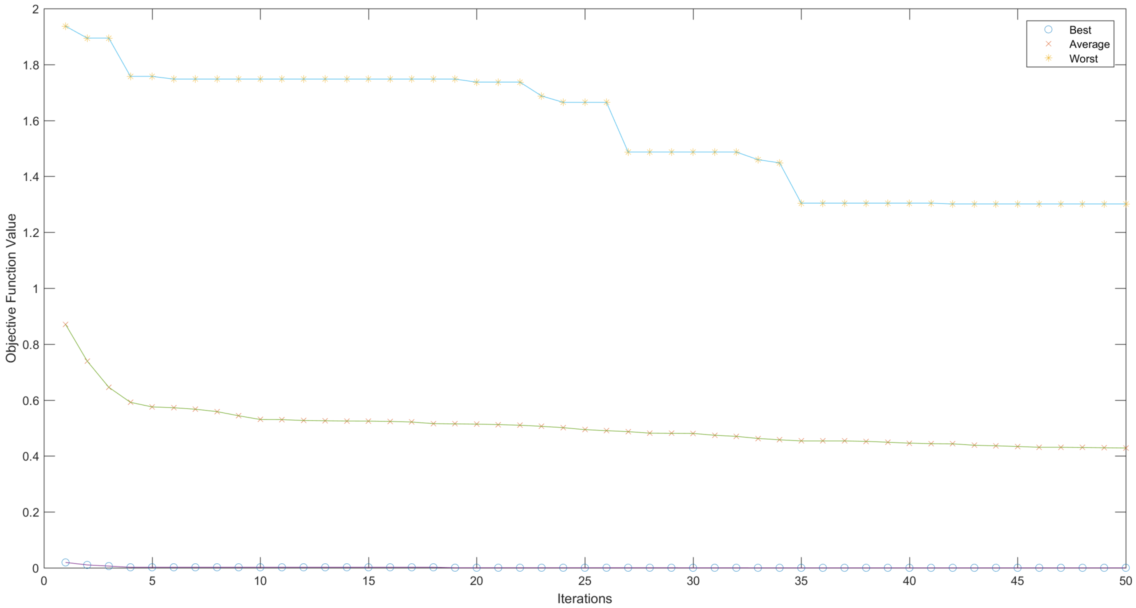
The constriction factor version of the PSO algorithm was used. The values are included in the MATLAB Code.

We compute gbest, pbest , best, worst and average objective function Values using MATLAB.

$$\text{The local minimum is found at } X^* = \begin{bmatrix}-0.0194\\-0.0307\end{bmatrix}$$

The value of the function at this minimum is 0.00024341.

The Plot of the best, worst and average objective function values is included below:-



MATLAB CODE

```
% Minimize Griewank PSO Algorithm

D=2;                    % Dimension of Griewank Function

x_min = -5;             % Bound Position and Velocities
x_max = 5;
v_min = -1;
v_max = 1;

N_iterations = 50;                %Number of iterations
d = 100;                          % Number of Particles

% Initialization of PSO parameters ( Constriction-factor version )

K=0.729;
c1=2;
c2=2.1;

best=zeros(N_iterations,1);
average=zeros(N_iterations,1);
worst=zeros(N_iterations,1);
F_best=zeros(N_iterations,1);

% Initialize the position, velocities, pbest & gbest of particles

p_pos = x_min + (x_max-x_min).*rand(d,D);
p_vel = v_min + (v_max-v_min).*rand(d,D);

p_best = p_pos;

for k=1:d
    F_best(k)=griewank(p_pos(k,:));
end

[g_best_value,g_best_index]=min(F_best);
g_best=p_pos(g_best_index,:);

% PSO Algorithm with the Constriction Factor

for k=1:N_iterations
    for i=1:d
```

```
        r=rand(1,2);
        s=rand(1,2);
        p_vel(i,:) = 0.729*(p_vel(i,:) + c1*r.*(p_best(i,:)-p_pos(i,:)) + c2*s.*(g_best-p_pos(

        p_vel(i,:) = min(v_max,max(v_min,p_vel(i,:)));    % Clamp Velocities

        p_pos(i) = p_pos(i) + p_vel(i);                 % Updating Positions of PS

        if p_pos(i,1) > 5                              % Restrict Positions
            p_pos(i,1)=5;
        end
        if p_pos(i,1) < -5
            p_pos(i,1)=-5;
        end
        if p_pos(i,2) >5
            p_pos(i,2)=5;
        end
        if p_pos(i,2) < -5
            p_pos(i,2)=-5;
        end
    end

    for i=1:d                                        % Compute pbest & gbest
        F_new(i)=griewank(p_pos(i,:));               % Computing Function Values using p_best pop

        if F_new(i)<F_best(i)
            F_best(i)=F_new(i);
            p_best(i,:)=p_pos(i,:);
        end
    end

    [g_best_value,g_best_index]=min(F_best);
    g_best=p_pos(g_best_index,:);                   % Compute g_best

    best(k)=min(F_best);                       % Compute best,worst and average objective function
    average(k)=mean(F_best);                           % Every Iteration
    worst(k)=max(F_best);

end

disp('The Minimizer Point is');
disp(g_best);                               % Plot the Minimum Point
disp('The Objective function value at the Minimum Point is');
disp(g_best_value);

figure;                                     % Plot Values
x=1:N_iterations;
plot(x,best,'o',x,average,'x',x,worst,'*');
hold on;
plot(x,[best average worst]);
hold off;
legend('Best','Average','Worst');
xlabel('Iterations');
ylabel('Objective Function Value');
```

Answer 5

The MATLAB Code is include after the Answer.
The Griewank function is Maximized by the PSO algorithm.
The Dimension is 2. The number of Particles in the Swarm is set to 100.

The value of the coordinates for each particle is clipped between [-5,5].

The value of the velocities for each particle is clipped between [-1,1].

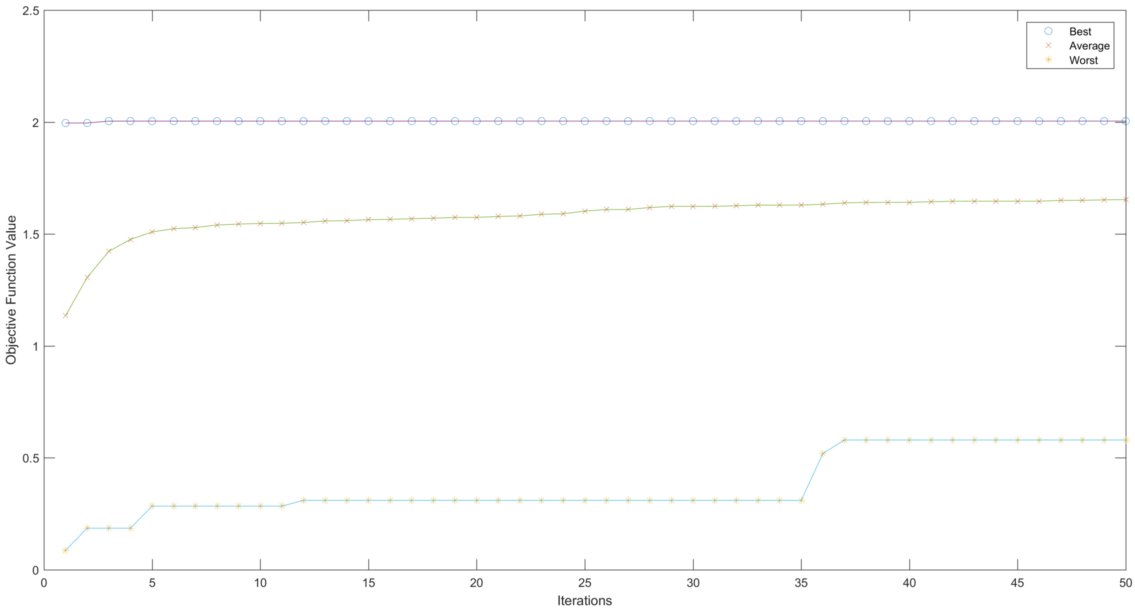
The constriction factor version of the PSO algorithm was used. The values are included in the MATLAB Code.

We compute gbest, pbest , best, worst and average objective function Values using MATLAB.

$$\text{The local minimum is found at } X^* = \begin{bmatrix} 0.0340 \\ 4.4874 \end{bmatrix}$$

The value of the function at this minimum is 2.0045.

The Plot of the best, worst and average objective function values is included below:-



MATLAB CODE

```
% Minimize Griewank PSO Algorithm

D=2;                    % Dimension of Griewank Function

x_min = -5;             % Bound Position and Velocities
x_max = 5;
v_min = -1;
v_max = 1;

N_iterations = 50;                  %Number of iterations
d = 100;                             % Number of Particles

% Initialization of PSO parameters ( Constriction-factor version )

K=0.729;
c1=2;
c2=2.1;

best=zeros(N_iterations,1);
average=zeros(N_iterations,1);
worst=zeros(N_iterations,1);
F_best=zeros(N_iterations,1);

% Initialize the position, velocities, pbest & gbest of particles

p_pos = x_min + (x_max-x_min).*rand(d,D);
p_vel = v_min + (v_max-v_min).*rand(d,D);

p_best = p_pos;

for k=1:d
    F_best(k)=griewank(p_pos(k,:));
```

```
end

[g_best_value, g_best_index]=max(F_best);
g_best=p_pos(g_best_index,:);

% PSO Algorithm with the Constriction Factor

for k=1:N_iterations
    for i=1:d
        r=rand(1,2);
        s=rand(1,2);
        p_vel(i,:) = 0.729*(p_vel(i,:) + c1*r.*(p_best(i,:)-p_pos(i,:)) + c2*s.*(g_best-p_pos(

        p_vel(i,:) = min(v_max,max(v_min,p_vel(i,:)));   % Clamp Velocities

        p_pos(i) = p_pos(i) + p_vel(i);              % Updating Positions of PS

        if p_pos(i,1) > 5                            % Restrict Positions
            p_pos(i,1)=5;
        end
        if p_pos(i,1) < -5
            p_pos(i,1)=-5;
        end
        if p_pos(i,2) >5
            p_pos(i,2)=5;
        end
        if p_pos(i,2) < -5
            p_pos(i,2)=-5;
        end
    end

    for i=1:d                                        % Compute pbest & gbest
        F_new(i)=griewank(p_pos(i,:));               % Computing Function Values using p_best pop

        if F_new(i)>F_best(i)
            F_best(i)=F_new(i);
            p_best(i,:)=p_pos(i,:);
        end
    end

    [g_best_value, g_best_index]=max(F_best);
    g_best=p_pos(g_best_index,:);                    % Compute g_best

    best(k)=max(F_best);                  % Compute best, worst and average objective function
    average(k)=mean(F_best);                         % Every Iteration
    worst(k)=min(F_best);

end

disp('The Maximizer Point is');
disp(g_best);                             % Plot the Maximum Point
disp('The Objective function value at the Maximum Point is');
disp(g_best_value);

figure;                                   % Plot Values
x=1:N_iterations;
plot(x,best,'o',x,average,'x',x,worst,'*');
hold on;
plot(x,[best average worst]);
hold off;
legend('Best','Average','Worst');
xlabel('Iterations');
ylabel('Objective Function Value');
```

THE END