Project- EE 840.3 Mathematical Methods in Engineering

Submitted By:

Mahdi Md. Mostafa

Student ID: 11270121

NSID: mmm874

Submitted to:

Associate Professor Dr. N. Chowdhury

Department of Electrical and Computer Engineering

University of Saskatchewan

University of Saskatchewan

Using particles swarm Optimization (PSO) for the optimal positioning of Wind Farm Layout for the reduction of wake loss

Introduction

Due to the need to maintain consistent, environment-friendly, and economic power sources, wind power energy has become popular. A wind turbine is a mechanical device that converts wind kinetic energy into electrical energy. Wind turbines are combined in the wind farm to produce a larger amount of electrical energy. The design of the placement of the individual wind turbine within the farm has been a major challenge in wind energy production. This project presents an optimization method using the particle swarm optimization (PSO) for the optimal design of farm layout for the optimization of power output.

Farm Layout Design Problem

The optimal wind turbine positioning is a 2D problem that involves determining the (x,y) coordinate of each turbine in the farm for optimal power production and minimum wake losses (Charhouni, Sallaou, and Mansouri, 2019).

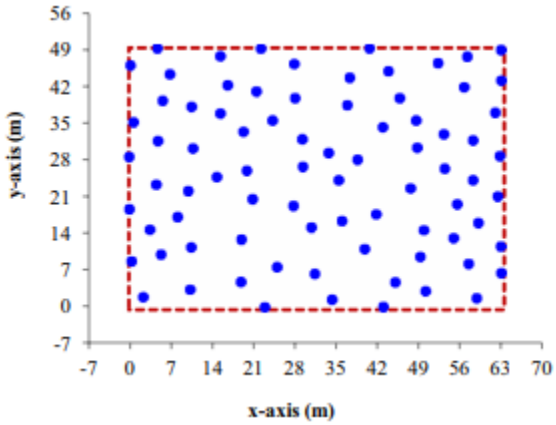


Figure 1:Wind turbine layout

Power model

To formulate the power production in a wind farm under the wake effect, there is a need to determine the power generated by each of the turbines. According to Charhouni, Sallaou, and Mansouri, (2019), the power produced by a wind turbine is estimated using equation 1.

$$P_t = \eta \frac{1}{2} \boldsymbol{\rho} A u^3 C_p \; - - - (1)$$

Where, $\eta = wind\ efficiency$

$\rho =$ Air density

$u = Air\ stream\ velocity$

$C_p =$ Power coefficient / efficiency factor

$$C_p = \frac{P_A}{\left(\frac{1}{2}\right)\rho A u^3} - - - (2)$$

Given, $\eta = 40\%, \rho = 1.2,\ r = 20m$

$$P_t = \frac{40}{100} \times \frac{1}{2} \times 1.2 \times \Pi \times u^3$$

$$P_t = 301\ u^3\ Watts$$

Hence, the power production by a wind turbine is given by:

$$P = 0.3 \times u_o^3 - - - (3)$$

Equation 3 shows that the power produced by a wind turbine on the farm is directly proportional to the cube of available free stream velocity. However, it is important to note that this is an ideal situation, the wake effect makes the equation to change to equation 4.

$$P = 0.3 \times (u_o - u_d)^3 - - - (4)$$

Where, $u_d = velocity\ deficit$

Wake Effect

The wind speed passing through the wind farm is reduced due to the wake effect which is based on the principle of conservation of momentum. Wake effect is the phenomenon that occurs when a wind turbine casts a wind shade resulting in less turbulent and slow wind speed behind the wind turbine.

The placement of wind turbines in the wind farm is primarily dependent on the wake effects within the farm and between the turbines. Wind turbines are spaced vertically and horizontally in the farm to avoid velocity deficit due to the wake effect.

The pictorial view of the wake effect is described as described by Charhouni, Sallaou, and Mansouri, (2019) is shown in figure 2.

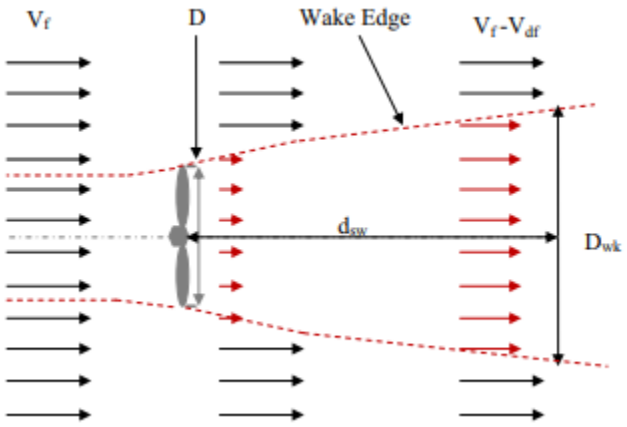


Figure 2:Illustration of wake effect

The velocity deficit factor is the ratio by which the free stream velocity is reduced due to the wake effect which is given by equation 5.

$$u = (1 - u_{df}) \times u_o - - - (5)$$

$$u_{df} = \frac{2a}{1 + (\frac{ax}{r_i})^2} - - - (6)$$

Hence,

$$u = u_{\circ}\left(1 - \frac{2a}{1 + (\frac{ax}{r_i})^2}\right) - -(7)$$

Where $a = induction\ factor$. which is given by:

$$a = \frac{0.5}{\ln(\frac{H}{z_o})} - -(8)$$

$H = hub\ height, z_o = surface\ roughness$

$r_i = vertical\ spacing = r_d + ax - -(9)$

The wake effect due to the vertical spacing of the wind turbine is described in figure 3

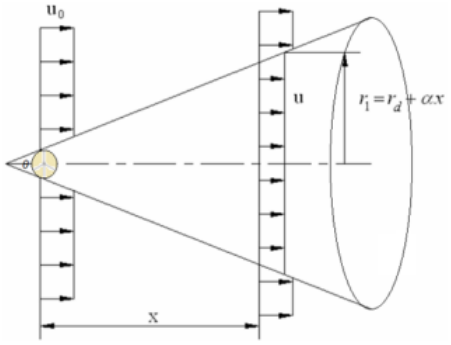


Figure 3: Illustration of the vertical spacing

Cost model

The main goal of this optimization is to ensure maximum energy is generated within the wind farm at the lowest possible cost. Hence, cost modeling is essential, the cost modeling is the process of estimating the total value of cost needed to run the turbine for a particular period. The cost equation defines the total cost of running N turbines per unit time. The cost model was obtained from Shakoor et al., (2015), which is given the equation below

$$Cost = N_t \left[\frac{2}{3} + \frac{1}{3} e^{-0.00174Nt2}\right] - - - (10)$$

Where *Nt* =Number of a wind turbine in the farm

The cost model can be used for the optimization problem by incorporating it with the power model as shown in equation 7. Hence, the optimization problem is aimed at minimizing the objective function.

$$objective = minimize\left(\frac{cost}{total\ power}\right) - -(11)$$

Assumptions

The following assumptions are made for the simplicity of the model:

1. The freestream velocity for the farm, u_o is 12 m/s
2. The air density is constant and it's equal to 1.2253 kg/m3
3. The farm layout is running for a known number of turbines

Constraint

The minimum distance between any two turbines is assumed to be 200m

Optimization

The particle swarm optimization (PSO) algorithm is proposed for the optimization problem. Particle swarm optimization is a global optimization algorithm that uses evolutionary computation problem to search for the best solution iteratively (El-Shorbagy and Hassanien, 2018). PSO has been applied to many optimization problems which include the constraint optimization problem, multi-objective optimization, and stochastic optimization problem.

PSO search algorithm is inspired by the behavior of bees in search of food in an open field. The algorithm search for the best solution within a search space by observing the best result from individual solutions.

The mathematical formulation of the PSO is described below:

parameters

$x_i^t = particle\ i\ at\ iteration\ t$

$v_i^t = velocity\ of\ particle\ i\ at\ iteration\ t$

$xc_i^t = the\ cost\ of\ particle\ i\ at\ iteration\ t$

$bx_i^t = the\ best\ position\ of\ particle\ i\ at\ iteration\ t$

$gx^t = the\ global\ best\ position\ for\ all\ particles\ at\ iteration\ t$

$w = damping\ coefficient$

$c_1, c_2 = learning\ factor$

$r_1, r_2 = random\ number$

Updating velocity

$$v_i^{t+1} = w \times v_i^t + c_1 \times r_1(bx_i^t - x_i^t) + c_2 \times r_2 \times (gx^t - x_i^t) ----(12)$$

Updating position

$$x_i^{t+1} = x_i^t + v_i^{t+1} \quad -------(13)$$

The optimization procedure is designed to compute the optimal coordinate point for each of the wind turbines in the wind farm for the minimization of cost and maximization of power. Equation 11 is used the objective function. The expected optimization result is the coordinate values for x and y for each of the turbines on the farm.

Optimization Procedure

To define the optimization procedure, the following key terms are defined as follows:

1. Particle: A particle represents a solution within the search space. Regarding the optimization problem, the particle position is a set of coordinates that defines the position of a turbine in the farm. Example of an individual is [(x1,y1),[x2,y2]….(xn,yn)] for n turbines. While the particle cost is the objective value of the particle position.
2. Population: A set of particles
3. Particle velocity: This is the rate at which the position of a particle change
4. Velocity updating: This is the process of changing the particle velocity of a particular particle based on its best solution and the global solution using equation (12)
5. Position updating: This is the process of changing the particle position base on its velocity using equation (11)
6. Global best cost: The best objective value attains during the search.
7. Global best position: The position that gave the global best cost.

The following steps described the optimization procedures:

Step 1: Create N random population of particles (N is the population size)

Step 2: Compute the objective value for each of the particles in the solution and determine the global best position and cost

Step 3: Start iterations

Step 4: Loop for each particle in the population

Step 5: Update the velocity of the particle using equation 12

Step 6: Update the position of the particle using equation (13)

Step 7: Compute the objective value of the particle using the new position

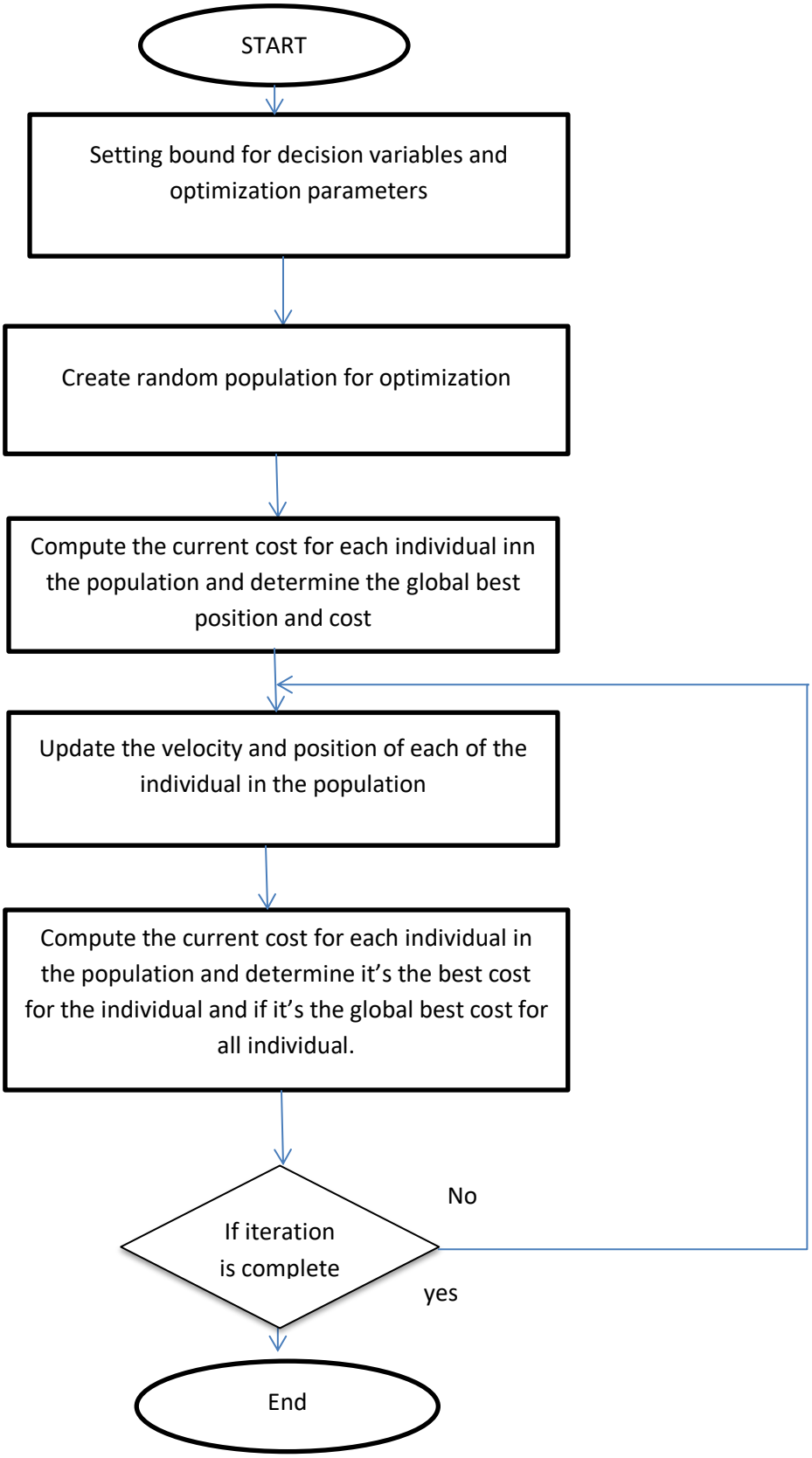
Step 8: Determine if it's the best position with the objective value

Step 9: Determine if it's the global position with the objective value

Step 10: Next particle

Step 11: Next iteration

The flow chart for the PSO algorithm is described in the next page.


START
Setting bound for decision variables and optimization parameters
Create random population for optimization
Compute the current cost for each individual inn the population and determine the global best position and cost
Update the velocity and position of each of the individual in the population
Compute the current cost for each individual in the population and determine it's the best cost for the individual and if it's the global best cost for all individual.
If iteration is complete
No
yes
End

Parameters and characteristics of the wind farm

Table 1: Farm layout parameters

Number of turbines	10
Hub height	60m
Rotor radius(rd)	40m
Wind farm area(km2)	1km x 1km
Length of surface roughness(m)	0.3
Free stream velocity(m/s)	12
Air density(kg/m2)	1.2253

PSO parameter values

The simulation was carried out with a population of 50 for 100 iterations. The constraint tolerance and functional tolerance was set to 1e-100 for better optimization. The simulation was carried out on the Matlab Software.

Optimization Results

The simulation was carried in the Matlab software using the farm layout parameters and the PSO parameter values discussed above. Table 2 describes the adjustment of the position of the turbine within the iterations to maximize the total power production generated in the wind farm. It also shows that the cost value is constant all through the optimization because the cost model only depends on the number of the turbine which is constant. However, the result was able to obtain an increment in the total power all during the iterations.

The coordinate of the optimal position of the turbine is shown in table 3 and figure 4

Table 2: Optimization Result

Iteration	Position of turbine **X**	**Y**	Power (Watts)	Cost
1	[417.022,19.367,326.65,71.974,950.176,0.402,811.859,770.239,959.434,187.6316]	[87.482,338.708,674.5640,357.062,138.355,477.862,888.386,859.651,245.518,381.081]	2.3754×10^{11}	16.657
20	[869.0575,623.315,0,82.3085,922.0610,1000,467.93906,467.603,0,363.590]	[783.118,1000,576.3477,1000,1000,0,0,485.826,793.572,871.785]	2.2845×10^{12}	16.657
40	[0,9.889,628.7658,1000,7.864,920.519,0,0,931.323,299.0935]	[793.806,562.927,1000,0,0,1000,263.249,1000,588.0180,0]	1.4860×10^{12}	16.657
60	[0,9.947,618.672,1000,6.8580,927.8719,0,0,935.14737,298.9236]	[797.062,576.232,1000,0,0,1000,271.966,1000,569.52818,0]	4.4030×10^{12}	16.657

80	[0,9.9472,618.6718,1000,6.8579,927.8727,0,0,935.145,298.9236]	[797.06,576.233,1000,0,0,1000,271.9673,1000,569.52627,0]	4.4319×10^{12}	16.657
100	[0,9.947,618.664,1000,6.857,927.877,0,0,935.150,298.923]	[797.0654,576.242,1000,0,0,1000,271.97,1000,569.5133,0]	4.4320×10^{12}	16.657
120	[0,9.947,618.618,1000,6.8525,927.912,0,0,935.168,298.922]	[797.080,576.3050,1000,0,0,1000,272.01423,1000,569.426753,0]	4.4320×10^{12}	16.657
140	[0,9.949,618.299,1000,6.8208,928.14,0,0,935.288,298.917]	[797.18,576.72,1000,0,0,1000,272.289,1000,568.84,0]	4.4332×10^{12}	16.657
160	[0,9.9617,616.158,1000,6.607,929.703,0,0,936.099,298.881]	[797.873907023007,579.546007493146,1000,0,0,1000,274.137740571891,1000,564.922448045321,0]	4.4406×10^{12}	16.657
180	[970.24,199.2480,427.784,749.55,0,0,1000,843.9151432,0,0]	[370.2355,124.907,1000,726.7891,213.91,0,885.7310,0,1000,752.104]	4.1965×10^{12}	16.657
200	[0,189.546,443.106,717.3531,0,0,1000,718.314,0.484,0]	[418.65,309.293,692.55,886.805,209.461,0,916.497,132.696,1000,750.144]	4.0299×10^{12}	16.657

Table 3: The optimal position of the turbines in the wind farm

From 200th iteration of table 2 the values of X and Y coordinates are given in the table 3. First row is for X and 2nd Row is for Y.

0	189.546	443.106	717.353	0	0	1000	718.314	0.484	0
418.65	309.293	692.55	886.805	209.461	0	916.497	132.696	1000	750.144

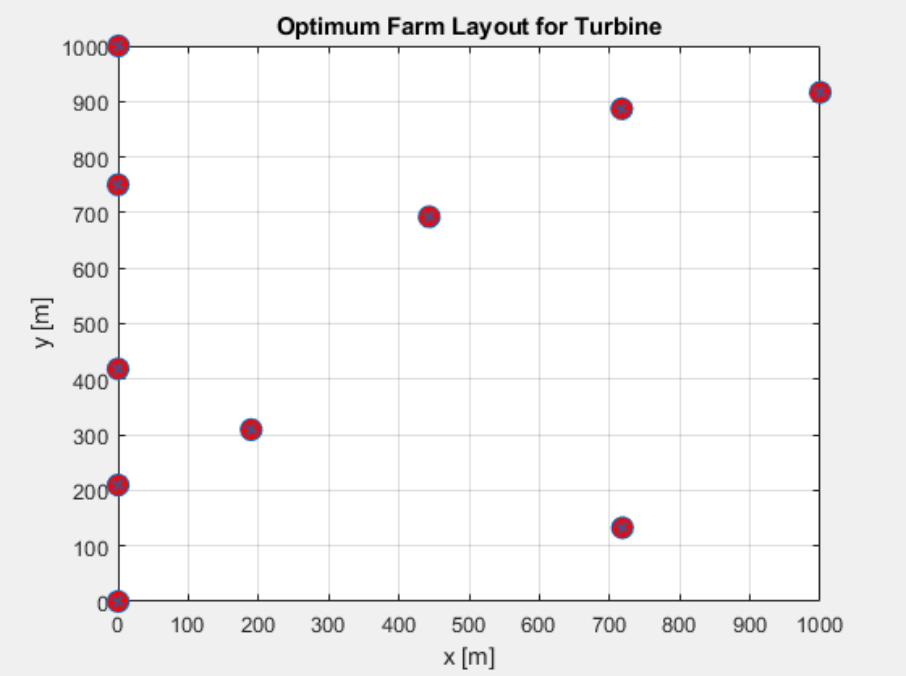


Figure 4: Optimum farm layout

Conclusion

This project was able to formulate the wind power production model in a wind farm taking into account of wake effect. Since the wind power production is affected by the wake effect within the wind farm, hence an optimization procedure was developed to determine the optimal position of each turbine in the farm to minimize wake loss. The result obtained shows that the optimization technique was able to increase the wind power production.

References

M.A. El-Shorbagy;Aboul Ella Hassanien. (2018). Particle Swarm Optimization from Theory to Applications. *International Journal of Rough Sets and Data Analysis, 5*(2).

Naima Charhouni;Mohammed Sallaou;Khalifa Mansouri. (2019). Realistic wind farm design layout optimization with diferent wind turbines types. *International Journal of Energy and Environmental Engineering*.

Nicholas F. Baker;1 Andrew P.J. Stanley;Jared J. Thomas;1 Andrew Ning;Katherine Dykes. (2019). *Best Practices for Wake Model and Optimization Algorithm Selection in Wind Farm Layout Optimization.* California.

Rabia Shakoor;Mohammad Yusri Hassan;Abdur Raheem;Nadia Rasheed. (2015). The Modelling of Wind Farm Layout Optimization for the Reduction of Wake Losses. *Indian Journal of Science and Technology, 8*(17).

Appendix

Maincode.m

```
clc
clear
close

rng(1) %reproductivity

global state_record
state_record=[];
%using PSO parameters
pop=50;  %population size
maxit=200; %maximum iteration

%plant parameters
H=60;    % Hub height
rd=40;   %Rotor radius
Cy=0.88; %Thrust coefficient
uo=12; %free strea velocity
zo=0.3; %surface roughness
Nt=10; %Number of turbine

fun =@(x) objfcn(x); %objective function

lb=zeros(1,2*Nt); %lower bound for optimization variables
ub=1e3*ones(1,2*Nt); %upper bound for parameter

%using PSO Algorithm
options =
optimoptions('particleswarm','SwarmSize',pop,'Display','iter','MaxStallIterat
ions',maxit,...
    'MaxIterations',maxit,'FunctionTolerance',1e-100,
'OutputFcn',@psooutfun);

variables = particleswarm(fun,2*Nt,lb,ub,options);
clear psooutfun
record = psooutfun();

%induction factor
a=0.5/log(H/zo);

%extract the optimized x and y coordinates
x=variables(1:Nt);  %x coodinates
y=variables(Nt+1:end); %y coordinates

%plot the graph
figure(1)
plot(x,y,'o', 'MarkerSize',10,'MarkerFaceColor','r')
xlabel('x [m]')
ylabel('y [m]')
grid on
title('Optimum Farm Layout for Turbine')
```

```
%function to record iteration output
function stop= psooutfun(options,state)
    global state_record

    if isempty(state_record) || length(state_record)>203
      state_record = struct('position',{},'power',{},'cost',{});
    end
    if nargin == 0
      state = state_record;
      options = [];
      stop = [];
    else
     [power,cost]=objective(options.bestx);
     %state_record = vertcat(state_record, [power,cost]);
     state_record(end+1) =
struct('position',options.bestx,'power',power,'cost',cost);
     stop = false;
     state=state_record;
    end

end
```

constraint.m

```
function [x,bound] = constraint(x)

%Constaint functions
Nt=10;   %Number of turbine

%extract y coordinate
y=x(Nt+1:end);

%initial the zeros distance
dist=zeros(Nt,Nt);
s=200; %minimum spacing of 500m
bound=0;

%calculate the distance between the turbine and check for minimum spacing
for i=1:Nt
      for j=1:Nt
          dist(i,j)=sqrt((x(i)-x(j))^2 + (y(i)-y(j))^2);
          if dist(i,j)<s && i~=j
             bound=1;
          end
      end
end

end
```

objfcn.m

```
function  objective= objfcn(x)

%Objective function

H=60;      %Hub height
rd=40;     %Rotor radius
Cy=0.88;   %Thrust coefficient
uo=12;     %free strea velocity
zo=0.3;    %surface roughness
Nt=10;     %Number of turbine


%induction factor
a=0.5/log(H/zo);

%compute the cost value
cost=Nt*(2/3 + (1/3)*exp(-0.00174*(Nt.^2)));

total_power=0;

%ccompute the reduction in the velocity
for i=1:Nt

    xi=x(i);
    ri=rd+a*xi;
    us(i)=uo*(1-2*a/((1+a*xi)/ri).^2);

    k(i)=(1-us(i)/uo).^2;
end

ui=uo*sqrt(sum(k));

%compute the total power
for i=1:Nt

    power=0.3*ui.^3;
    total_power=total_power+power;

end

%check for constraint
[~,bound] = constraint(x);

%set penalty for value out of the constraints
if bound ==1
     objective=100e3;
else
    objective=cost/ total_power;
end

end
```

objective.m

```
function  [total_power,cost]= objective(x)

%Objective function

H=60;      %Hub height
rd=40;     %Rotor radius
Cy=0.88;   %Thrust coefficient
uo=12;     %free strea velocity
zo=0.3;    %surface roughness
Nt=20;     %Number of turbine

%induction factor
a=0.5/log(H/zo);

%compute the cost value
cost=Nt*(2/3 + (1/3)*exp(-0.00174*(Nt.^2)));

total_power=0;

%ccompute the reduction in the velocity
for i=1:Nt

    xi=x(i);
    ri=rd+a*xi;
    us(i)=uo*(1-2*a/((1+a*xi)/ri).^2);

    k(i)=(1-us(i)/uo).^2;
end

ui=uo*sqrt(sum(k));

%compute the total power
for i=1:Nt
    power=0.3*ui.^3;
    total_power=total_power+power;
end


end
```