**Appendix:**

**Matlab Code**

%Design Optimization of Wind Turbine Placement using SQP and BFGS with a

%Trust Region – Scaled to 24 Turbines

clear all

thetaw=pi/1.01; alpha=1/3; R=50; k=.1; U=12; A=pi\*R^2; rhoa=1.225; D=2\*R; %Initial conditions for theorized wind farm location

Cp = 4\*alpha\*(1-alpha)^2;

syms l1 l2 l3 l4 l5 l6 l7 l8 l9 l10 l11 l12 l13 l14 l15 l16 l17 l18 l19 l20 l21 l22 l23 l24 l25 l26 l27 l28 l29 l30 l31 l32 l33 l34 l35 l36 l37 l38 l39 l40 l41 l42 l43 l44 l45 l46 l47 l48 %symbolic placeholders for calculating the power function symbolically

l\_sym=[l1 l2 l3 l4 l5 l6 l7 l8 l9 l10 l11 l12 l13 l14 l15 l16 l17 l18 l19 l20 l21 l22 l23 l24 l25 l26 l27 l28 l29 l30 l31 l32 l33 l34 l35 l36 l37 l38 l39 l40 l41 l42 l43 l44 l45 l46 l47 l48];

%power function loop to calclate power equation

for i=1:24

for j=1:24

theta\_ij(i)=sec(dot(l\_sym([2\*i-1,2\*i]),l\_sym([2\*j-1,2\*j])))/(norm(l\_sym([2\*i-1,2\*i])\*norm(l\_sym([2\*j-1,2\*j]))));

r\_ij(i)=norm(l\_sym([2\*i-1,2\*i])-l\_sym([2\*j-1,2\*j]))\*sin(abs(theta\_ij(i)-thetaw));

d\_ij(i)=norm(l\_sym([2\*i-1,2\*i])-l\_sym([2\*j-1,2\*j]))\*cos(abs(theta\_ij(i)-thetaw));

du\_ij(i)= 2\*alpha\*(R/(R+k\*d\_ij(i)))^2\*exp(-(r\_ij(i)/(R+k\*d\_ij(i)))^2);

dubar\_ij(i)=sqrt(sum(du\_ij));

ubar\_ij(i)=U\*(1-dubar\_ij(i));

P=0.5\*rhoa\*A\*Cp\*(ubar\_ij(i))^3; %Solving for the Power equation in terms of l for every l value

end

end

lk=[1.1;1.2;6.1;1.4;11.1;1.5;16.2;1.3;21.4;1.1;26.7;1.42;1.1;6.2;6.2;6.3;11.6;6.1;16.3;6.3;21.5;6;26.3;6.1;1.3;11.1;6.2;11.7;11.2;11.5;16.4;11.2;21.1;11.2;26.1;11.4;1.4;16.4;6.4;16.4;11;16.5;16.2;16.3;21.3;16.5;26.7;16.2]; %initial point

lk=1.5\*R.\*lk; %original turbine placement location non ideal - 1.5 radii times the grid

l\_original = lk;

l\_old = 1e1\*ones(size(lk));

rho=1;

Bk=eye(48);

while norm([lk-l\_old],2)>0.001

% initial guess is lk

l0 = lk;

A = [];

% constraint1(l, lk);

b = [];

Aeq = []; beq = [];

lb=l0-D\*10;

ub=l0+D\*10;

g\_sym=symbolicgradient(l\_old);

g = cal\_g(g\_sym,lk);

[l, fval, exitflag, ~, lambda] = fmincon(@(l)convexsub(l,g,Bk,lk),l0,A,b,...

Aeq,beq,lb,ub,@(l)noncon(l,lk,rho)); %insert fmin con function

nc=sym\_nc(l, lk, rho);

ncLk=cal\_nc(nc, l ,lk);

lambda = lambda.ineqnonlin;

pLk = cal\_p(P,lk);

pL = cal\_pL(P,l);

% do trust region

alpha = (pLk-pL)/(convexsub(l\_old,g,Bk,lk)-pL);

if alpha <= 0.2

lk = l;

l\_old=lk;

rho = 1.1\* rho;

Bk=laghessian(lambda,l,lk, g, ncLk);

display(alpha);

else

l\_old = l;

rho = 0.5\* rho;

display(alpha);

end

end

%parse x y data

for i = 1:24

x(i) = lk(2 \* i - 1);

y(i) = lk(2 \* i);

xx(i) = l\_original(2 \* i - 1);

yy(i) = l\_original(2 \* i);

end

table1 = table(l\_original,lk);

figure

hold on

scatter(x,y,'b','marker','+')

scatter(xx,yy,'r','marker','o')

line([x; xx], [y; yy],'Color', 'k', 'LineStyle', '--');

title('Optimized Turbine Locations - Wind Direction - 180\* Wind Speed - 12 m/s')

xlabel('x (m)')

ylabel('y (m)')

legend('Optimized Locations','Original Locations')

grid on

grid minor

hold off

function g\_sym= symbolicgradient(l\_old)

thetaw=pi/1.01; alpha=1/3; R=50; k=2; U=30; A=pi\*R^2; rhoa=1.225; Cp = 4\*alpha\*(1-alpha)^2; % Place Holders

syms l1 l2 l3 l4 l5 l6 l7 l8 l9 l10 l11 l12 l13 l14 l15 l16 l17 l18 l19 l20 l21 l22 l23 l24 l25 l26 l27 l28 l29 l30 l31 l32 l33 l34 l35 l36 l37 l38 l39 l40 l41 l42 l43 l44 l45 l46 l47 l48

l\_old=[l1 l2 l3 l4 l5 l6 l7 l8 l9 l10 l11 l12 l13 l14 l15 l16 l17 l18 l19 l20 l21 l22 l23 l24 l25 l26 l27 l28 l29 l30 l31 l32 l33 l34 l35 l36 l37 l38 l39 l40 l41 l42 l43 l44 l45 l46 l47 l48];

for i=1:24

for j=1:24

theta\_ij(i)=sec(dot(l\_old([2\*i-1,2\*i]),l\_old([2\*j-1,2\*j])))/((norm(l\_old([2\*i-1,2\*i])))\*norm(((l\_old([2\*j-1,2\*j])))));

r\_ij(i)=norm((((l\_old([2\*i-1,2\*i]))-l\_old([2\*j-1,2\*j]))))\*sin(abs(theta\_ij(i)-thetaw));

d\_ij(i)=norm(((l\_old([2\*i-1,2\*i])-l\_old([2\*j-1,2\*j]))))\*cos(abs(theta\_ij(i)-thetaw));

du\_ij(i)= 2\*alpha\*(R/(R+k\*d\_ij(i)))^2\*exp(-(r\_ij(i)/(R+k\*d\_ij(i)))^2);

dubar\_ij(i)=sqrt(sum(du\_ij));

ubar\_ij(i)=U\*(1-dubar\_ij(i));

P=0.5\*rhoa\*A\*Cp\*(ubar\_ij(i))^3; %Solving for the Power equation in terms of l for every l value

end

end

g\_sym = gradient(P);

end

function g = cal\_g(g\_sym, lk)

l1=lk(1); l3=lk(3); l5=lk(5); l7=lk(7); l9=lk(9); l11=lk(11); l13=lk(13); l15=lk(15); l17=lk(17);

l2=lk(2); l4=lk(4); l6=lk(6); l8=lk(8); l10=lk(10); l12=lk(12); l14=lk(14); l16=lk(16); l18=lk(18);

l19=lk(19); l20=lk(20); l21=lk(21); l22=lk(22); l23=lk(23); l24=lk(24); l25=lk(25); l26=lk(26); l27=lk(27);

l28=lk(28); l29=lk(29); l30=lk(30); l31=lk(31); l32=lk(32); l33=lk(33); l34=lk(34); l35=lk(35); l36=lk(36);

l37=lk(37); l38=lk(38); l39=lk(39); l40=lk(40); l41=lk(41); l42=lk(42); l43=lk(43); l44=lk(44); l45=lk(45);

l46=lk(46); l47=lk(47); l48=lk(48);

g = eval(g\_sym);

end

function pL = cal\_pL(P,l)

l1=l(1); l3=l(3); l5=l(5); l7=l(7); l9=l(9); l11=l(11); l13=l(13); l15=l(15); l17=l(17);

l2=l(2); l4=l(4); l6=l(6); l8=l(8); l10=l(10); l12=l(12); l14=l(14); l16=l(16); l18=l(18);

l19=l(19); l20=l(20); l21=l(21); l22=l(22); l23=l(23); l24=l(24); l25=l(25); l26=l(26); l27=l(27);

l28=l(28); l29=l(29); l30=l(30); l31=l(31); l32=l(32); l33=l(33); l34=l(34); l35=l(35); l36=l(36);

l37=l(37); l38=l(38); l39=l(39); l40=l(40); l41=l(41); l42=l(42); l43=l(43); l44=l(44); l45=l(45);

l46=l(46); l47=l(47); l48=l(48);

pL = eval(P);

end

function [pLk] = cal\_p(P, lk)

l1=lk(1); l3=lk(3); l5=lk(5); l7=lk(7); l9=lk(9); l11=lk(11); l13=lk(13); l15=lk(15); l17=lk(17);

l2=lk(2); l4=lk(4); l6=lk(6); l8=lk(8); l10=lk(10); l12=lk(12); l14=lk(14); l16=lk(16); l18=lk(18);

l19=lk(19); l20=lk(20); l21=lk(21); l22=lk(22); l23=lk(23); l24=lk(24); l25=lk(25); l26=lk(26); l27=lk(27);

l28=lk(28); l29=lk(29); l30=lk(30); l31=lk(31); l32=lk(32); l33=lk(33); l34=lk(34); l35=lk(35); l36=lk(36);

l37=lk(37); l38=lk(38); l39=lk(39); l40=lk(40); l41=lk(41); l42=lk(42); l43=lk(43); l44=lk(44); l45=lk(45);

l46=lk(46); l47=lk(47); l48=lk(48);

pLk = eval(P);

end

function c1=constraint1(l,lk)

D=100;

for i=1:24

for j=1:24

c1(i)=-(lk([2\*i-1,2\*i])-lk([2\*j-1,2\*j]))'\*(l([2\*i-1,2\*i])-l([2\*j-1,2\*j]))+3\*D\*norm(lk([2\*i-1,2\*i])-lk([2\*j-1,2\*j])) ;

c1(i+12)=-(lk([2\*i-1,2\*i])-lk([2\*j-1,2\*j]))'\*(l([2\*i-1,2\*i])-l([2\*j-1,2\*j]))+3\*D\*norm(lk([2\*i-1,2\*i])-lk([2\*j-1,2\*j])) ;

end

end

end

function y = convexsub(l, g, Bk, lk)

% x: the variable

% g: current function gradient (column)

% B: BFGS app. hessian of the Lagrangian

% xk: current value of the variables

y = g'\*(l - lk) + (l - lk)'\*Bk\*(l - lk);

end

function GC=gradient\_constraints(nc, l, lk)

ncLk=cal\_nc(nc, l, lk);

GC=ncLk;

end

function Bk=laghessian(lambda, l ,lk, g, ncLk)

B=eye(48);

s=lk-l;

new\_g\_L = g + (lambda'\*ncLk)';

old\_g\_L = g + (lambda'\*ncLk)';

y=new\_g\_L - old\_g\_L;

if s'\*y >= 0.2\*s'\*B\*s

theta = 1;

else

theta = (0.8\*s'\*B\*s)/(s'\*B\*s-s'\*y);

end

ytil=theta\*y+(1-theta)\*B\*s;

Bk=B+(1/(ytil'\*s))\*(ytil\*ytil')-(1/(s'\*B\*s))\*(B\*s\*s'\*B);

end

function [c\_n, ceq] = noncon(l,lk,rho)

for i = 1:24

c(i) = norm(l([2\*i-1,2\*i])-lk([2\*i-1,2\*i]),2)-rho;

end

c1 = zeros(96,1);

count = 1;

D=100;

for i=1:23

for j=i+1:24

c1(count)=-(lk([2\*i-1,2\*i])-lk([2\*j-1,2\*j]))'\*(l([2\*i-1,2\*i])-l([2\*j-1,2\*j]))...

+3\*D\*norm(lk([2\*i-1,2\*i])-lk([2\*j-1,2\*j])) ;

count = count + 1;

end

end

c\_n = [c';c1];

ceq = [];

end

function nc = sym\_nc(lk,l,rho)

% rho = 1.1;

syms l1 l2 l3 l4 l5 l6 l7 l8 l9 l10 l11 l12 l13 l14 l15 l16 l17 l18 l19 l20 l21 l22 l23 l24 l25 l26 l27 l28 l29 l30 l31 l32 l33 l34 l35 l36 l37 l38 l39 l40 l41 l42 l43 l44 l45 l46 l47 l48

syms lk1 lk2 lk3 lk4 lk5 lk6 lk7 lk8 lk9 lk10 lk11 lk12 lk13 lk14 lk15 lk16 lk17 lk18 lk19 lk20 lk21 lk22 lk23 lk24 lk25 lk26 lk27 lk28 lk29 lk30 lk31 lk32 lk33 lk34 lk35 lk36 lk37 lk38 lk39 lk40 lk41 lk42 lk43 lk44 lk45 lk46 lk47 lk48

l=[l1 l2 l3 l4 l5 l6 l7 l8 l9 l10 l11 l12 l13 l14 l15 l16 l17 l18 l19 l20 l21 l22 l23 l24 l25 l26 l27 l28 l29 l30 l31 l32 l33 l34 l35 l36 l37 l38 l39 l40 l41 l42 l43 l44 l45 l46 l47 l48];

lk=[lk1 lk2 lk3 lk4 lk5 lk6 lk7 lk8 lk9 lk10 lk11 lk12 lk13 lk14 lk15 lk16 lk17 lk18 lk19 lk20 lk21 lk22 lk23 lk24 lk25 lk26 lk27 lk28 lk29 lk30 lk31 lk32 lk33 lk34 lk35 lk36 lk37 lk38 lk39 lk40 lk41 lk42 lk43 lk44 lk45 lk46 lk47 lk48];

l=l';

lk=lk';

nc\_sym= cell(24,1);

for i = 1:24

c(i) = norm(l(2\*i-1:2\*i,1)-lk(2\*i-1:2\*i,1),2)-rho;

% c(i+9)=norm(l([2\*i-1,2\*i])-lk([2\*i-1,2\*i]),2)-rho;

nc\_sym{i,1} = gradient(c(i),l');

end

% c = norm([l-lk],2)-rho;

% gcc = gradient(c,l);

D=100;

nc1\_sym=cell(24,24);

for i=1:23

for j=(i+1):24

c1(i,j)=-(lk([2\*i-1,2\*i])-lk([2\*j-1,2\*j]))'\*(l([2\*i-1,2\*i])-l([2\*j-1,2\*j]))...

+3\*D\*norm(lk([2\*i-1,2\*i])-lk([2\*j-1,2\*j])) ;

% c1(i+9)=-(lk([2\*i-1,2\*i])-lk([2\*j-1,2\*j]))'\*(l([2\*i-1,2\*i])-l([2\*j-1,2\*j]))...

% +5\*D\*norm(lk([2\*i-1,2\*i])-lk([2\*j-1,2\*j])) ;

nc1\_sym{i,j}=gradient(c1(i,j),l');

end

end

nc\_sym=nc\_sym';

nc1\_sym=nc1\_sym';

nc = [nc\_sym ; nc1\_sym];

end

function [ncLk] = cal\_nc(nc, l, lk)

lk1=lk(1); lk3=lk(3); lk5=lk(5); lk7=lk(7); lk9=lk(9); lk11=lk(11); lk13=lk(13); lk15=lk(15); lk17=lk(17);

lk2=lk(2); lk4=lk(4); lk6=lk(6); lk8=lk(8); lk10=lk(10); lk12=lk(12); lk14=lk(14); lk16=lk(16); lk18=lk(18);

lk19=lk(19); lk20=lk(20); lk21=lk(21); lk22=lk(22); lk23=lk(23); lk24=lk(24); lk25=lk(25); lk26=lk(26); lk27=lk(27);

lk28=lk(28); lk29=lk(29); lk30=lk(30); lk31=lk(31); lk32=lk(32); lk33=lk(33); lk34=lk(34); lk35=lk(35); lk36=lk(36);

lk37=lk(37); lk38=lk(38); lk39=lk(39); lk40=lk(40); lk41=lk(41); lk42=lk(42); lk43=lk(43); lk44=lk(44); lk45=lk(45);

lk46=lk(46); lk47=lk(47); lk48=lk(48);

l1=l(1); l3=l(3); l5=l(5); l7=l(7); l9=l(9); l11=l(11); l13=l(13); l15=l(15); l17=l(17);

l2=l(2); l4=l(4); l6=l(6); l8=l(8); l10=l(10); l12=l(12); l14=l(14); l16=l(16); l18=l(18);

l19=l(19); l20=l(20); l21=l(21); l22=l(22); l23=l(23); l24=l(24); l25=l(25); l26=l(26); l27=l(27);

l28=l(28); l29=l(29); l30=l(30); l31=l(31); l32=l(32); l33=l(33); l34=l(34); l35=l(35); l36=l(36);

l37=l(37); l38=l(38); l39=l(39); l40=l(40); l41=l(41); l42=l(42); l43=l(43); l44=l(44); l45=l(45);

l46=l(46); l47=l(47); l48=l(48);

c1 = nc(1,:);

c2 = nc(2:end,:)';

c1\_val = zeros(24,48);

num\_c = 23\*24/2;

c2\_val = zeros(num\_c,48);

for i = 1:24

c1\_val(i,:) = eval(c1{i})';

end

% c1\_val\_column = reshape(c1\_val',9\*18,1);

% c1\_val = c1\_val';

count = 1;

for i=1:23

for j=(i+1):24

c2\_val(count,:)=eval(c2{i,j})';

count = count + 1;

end

end

% c2\_val\_column = reshape(c2\_val',num\_c\*18,1);

% c2\_val = c2\_val';

ncLk=[c1\_val ; c2\_val];

end