clear all;

clc;

Pmax = [455,455,130,130,162,80,85,55,55,55]; % Maximal power output (MW)

Pmin = [150,150,20,20,25,20,25,10,10,10]; % Minimal power output (MW)

a = [1000,970,700,680,450,370,480,660,665,670]; % $/h

b = [16.19,17.26,16.6,16.5,19.7,22.26,27.74,25.92,27.27,27.79]; % $/MWh

c = [0.00048,0.00031,0.002,0.00211,0.00398,0.00712,0.00079,0.00413,0.00222,0.00173]; % $/MWh^2

SUH = [4500,5000,550,560,900,170,260,30,30,30]; % Start-up HOT cost ($)

SUC = [9000,10000,1100,1120,1800,340,520,60,60,60]; % Start-up COLD cost ($)

Tcold = [5,5,4,4,4,2,2,0.5,0.5,0.5]; % Cooling time unit

InitialTON = [8,8,0,0,0,0,0,0,0,0]; % Initial ON unit

InitialTOFF = [0,0,-5,-5,-6,-3,-3,-1,-1,-1]; % Initial OFF unit

MDT = [8,8,5,5,6,3,3,1,1,1]; % Min. Down Time (MIN OFF)

MUT = [8,8,5,5,6,3,3,1,1,1]; % Min. UP Time (MIN ON)

x0 = [1,1,0,0,0,0,0,0,0,0]; % Initial condition

DEMAND = [700; 750; 850; 950; 1000; 1100; 1150; 1200; 1300; 1400; 1450; 1500; 1400; 1300; 1200; 1050; 1000; 1100; 1200; 1400; 1300; 1100; 900; 800;];

Reserves = 0.1\*DEMAND;

[Y,h] = sort(Pmax,'descend');

SA = 10;

G = 10;

T = 24;

MaxIt = 50; % Maximal iteration

c1 = 0.18; % range of c1 is [0, 0.18]

c2 = 0.82; % range of c2 is [0.19, 1]

[Y,h] = sort(Pmax,'descend');

redfox = zeros(SA,G\*T);

ab = 1/sqrt(2)\*ones(2,G\*T);

for n = 1:SA

redfoxl = initial(G,T);

[redfox(n,:),redfoxTOFF(:,:,n)] = Verify(redfoxl,G,T,x0,MUT,MDT,InitialTON,InitialTOFF,Pmax,DEMAND,Reserves,h);

[fitness(n)] = ObjectFitness(redfox(n,:),G,T,x0,MDT,Pmax,Pmin,DEMAND,a,b,c,SUH,SUC,Tcold,redfoxTOFF(:,:,n));

end

Gmbest = redfox(1,:);

Gfitness = fitness(1);

for n = 1:SA

Pmbest(n,:) = redfox(n,:);

Pfitness(n) = fitness(n);

if Gfitness > Pfitness(n)

Gmbest = Pmbest(n,:);

Gfitness = Pfitness(n);

end

end

for n = 1:SA

ab1(:,:,n) = ab;

end

Distance\_Fox\_Rat = zeros(SA, G\*T);

Gfitness\_values = zeros(1, MaxIt);

MinT = inf;

for k = 1:MaxIt

for n = 1:SA

[redfox(n,:),ab1(:,:,n)] = Update(redfox(n,:),G,T,Pmbest(n,:),Gmbest,fitness(n),Pfitness(n),Gfitness,c2,c1,k,MaxIt,ab1(:,:,n),MinT,n);

[redfox(n,:),redfoxTOFF(:,:,n)] = Verify(redfox(n,:),G,T,x0,MUT,MDT,InitialTON,InitialTOFF,Pmax,DEMAND,Reserves,h);

[fitness(n)] = ObjectFitness(redfox(n,:),G,T,x0,MDT,Pmax,Pmin,DEMAND,a,b,c,SUH,SUC,Tcold,redfoxTOFF(:,:,n));

if Pfitness(n) >= fitness(n)

Pmbest(n,:) = redfox(n,:);

Pfitness(n) = fitness(n);

end

if Gfitness > Pfitness(n)

Gmbest = Pmbest(n,:);

Gfitness = Pfitness(n);

end

end

Gfitness\_values(k) = Gfitness;

end

options=optimset('LargeScale','off','MaxPCGIter',50,'PrecondBandWidth',inf);

for t = 1:T

H = 2\*diag(c.\* Gmbest(G\*(t-1)+1:G\*t));

C = b.\* Gmbest(G\*(t-1)+1:G\*t);

Aeq = ones(1,10).\* Gmbest(G\*(t-1)+1:G\*t);

VLB = Pmin;

VUB = Pmax;

beq = DEMAND(t);

p0 = 0.5\*(VLB+VUB);

P = quadprog(H,C',[],[],Aeq,beq,VLB,VUB,p0,options);

Gptbest(:,t) = P'.\*Gmbest(G\*(t-1)+1:G\*t);

end

Gptbest'

disp('-----------------------------------------------------------------------');

resu = Gfitness;

fprintf('Total Cost is $ %.6f\n', resu);

disp('-----------------------------------------------------------------------');

figure;

plot(1:MaxIt, Gfitness\_values, 'LineWidth', 1.5);

xlabel('Generation');

ylabel('Total Generation Cost $');

title('Graph of Total Cost vs Generation');

function x = initial(G,T)

x = ones(1,G\*T);

ab = 1/sqrt(2)\*ones(2,G\*T);

x = rand(1,G\*T)<ab(2,:).^2;

#####

function [Fobject]=ObjectFitness(x,G,T,x0,MDT,Pmax,Pmin,PD,a,b,c,SUH,SUC,Tcold,TOFF)

options=optimset('LargeScale','off','MaxPCGIter',50,'PrecondBandWidth',inf);

for t = 1:T

H = [];

C = [];

Aeq = [];

VLB = [];

VUB = [];

i = 0;

for g = 1:G

if x(g+G\*(t-1)) == 1

i = i+1;

H(i,i) = 2\*c(g);

C(i) = b(g);

Aeq(i) = 1;

VLB(i) = Pmin(g);

VUB(i) = Pmax(g);

p0 = 0.5\*(VLB+VUB);

end

end

beq = PD(t);

H;

C';

Aeq;

beq;

VLB;

VUB;

p0;

[Pt,F] = quadprog(H,C',[],[],Aeq,beq,VLB,VUB,p0,options);

T

F

a

x(G\*(t-1)+1:G\*t)'

FC(t) = F+a\*x(G\*(t-1)+1:G\*t)';

FC(t)

%% Start-up cost

if t == 1

for g = 1:G

if x(g+G\*(t-1)) == 1&&x0(g) == 0

SU = SUH(g);

end

end

else

for g = 1:G

if x(g+G\*(t-1)) == 1&&x(g+G\*(t-2))==0

if TOFF(g,t-1) >= MDT(g)&& TOFF(g,t-1) <= MDT(g)+Tcold(g)

SU = SUH(g);

elseif TOFF(g,t-1) > MDT(g)+Tcold(g)

SU = SUC(g);

end

end

end

end

end

Fobject = SU + sum(FC); % Total cost

function [x,ab] = Update(x,G,T,Pbest,Gbest,fitness,Pfitness,Gfitness,c2,c1,k,ITERmax,ab,MinT,n)

r=rand;

p= rand;

dim = G\*T;

r1 = (fitness<Pfitness);

r2 = (fitness<Gfitness);

if r >= 0.5

if p > 0.18

Time(n,:) = rand(1, dim);

sps = x ./ Time(n,:);

Distance\_S\_Travel(n,:) = sps .\* Time(n,:);

Distance\_Fox\_Rat(n,:) = 0.5 .\* Distance\_S\_Travel(n,:);

tt = sum(Time(n,:)) / dim;

t = tt / 2;

Jump = 0.5 \* 9.81 \* t^2;

Z = Jump .\* c1/ITERmax;

elseif p <= 0.18

Time(n,:) = rand(1, dim);

sps = x ./ Time(n,:);

Distance\_S\_Travel(n,:) = sps .\* Time(n,:);

Distance\_Fox\_Rat(n,:) = 0.5 .\* Distance\_S\_Travel(n,:);

tt = sum(Time(n,:)) / dim;

t = tt / 2;

Jump = 0.5 \* 9.81 \* t^2;

Z = Jump .\* c2/ITERmax;

end

if MinT > tt

MinT = tt;

end

elseif r < 0.5

% Random walk

Z = c2-(c2-c1)\* k/ITERmax;

end

DZ = Z\*(r1\*(Pbest-x)+r2\*(Gbest-x));

for g = 1:G

for t = 1:T

ab(:,g+G\*(t-1)) = [cos(DZ(g+G\*(t-1))),-sin(DZ(g+G\*(t-1)));sin(DZ(g+G\*(t-1))),cos(DZ(g+G\*(t-1)))]\*ab(:,g+G\*(t-1));

x(g+G\*(t-1)) = (rand()<ab(2,g+G\*(t-1))^2);

end

end

function [x,TOFF] = Verify(x,G,T,x0,MUT,MDT,InitialTON,InitialTOFF,PMAX,PD,SR,h)

TON = zeros(G,T);

TOFF = zeros(G,T);

for t = 1:T

if t == 1

TON(:,t) = (InitialTON.\*x(G\*(t-1)+1:G\*t)+x(G\*(t-1)+1:G\*t))';

TOFF(:,t) = InitialTOFF'.\*~TON(:,t)+~TON(:,t);

else

flag = (TOFF(:,t-1) == 0&TON(:,t-1) < MUT')|(TON(:,t-1) == 0&TOFF(:,t-1) < MDT');

x(G\*(t-1)+1:G\*t) = x(G\*(t-2)+1:G\*(t-1)).\*flag'+x(G\*(t-1)+1:G\*t).\*~flag';

TON(:,t) = TON(:,t-1).\*x(G\*(t-1)+1:G\*t)'+x(G\*(t-1)+1:G\*t)';

TOFF(:,t) = TOFF(:,t-1).\*~TON(:,t)+~TON(:,t);

end

P = PMAX\*x(G\*(t-1)+1:G\*t)';

if P < PD(t)+SR(t)

for j = 1:G

g = h(j);

if x(g+G\*(t-1)) == 1

continue;

else

x(g+G\*(t-1)) = 1;

if TOFF(g,t) > MDT(g)

if t == 1

TON(g,t) = InitialTON(g)+1;

else

TON(g,t) = TON(g,t-1)+1;

end

TOFF(g,t) = 0;

else

l = t-TOFF(g,t)+1;

if l <= 0

l = 1;

end

x(g+G\*(l-1:t-1)) = 1;

if l == 1

TON(g,l) = InitialTON(g)+1;

TON(g,l+1:t) = TON(g,l)+[1:t-l];

else

TON(g,l:t) = TON(g,l-1)+[1:t-l+1];

end

TOFF(g,l:t) = 0;

end

P = PMAX\*x(G\*(t-1)+1:G\*t)';

if P >= (PD(t)+SR(t))

break;

end

end

end

end

for i = 1:G

g = h(G+1-i);

P1 = PMAX\*x(G\*(t-1)+1:G\*t)';

if x(g+G\*(t-1)) == 1

if P1-PMAX(g) >= PD(t)+SR(t)

if TON(g,t) > MUT(g)||(TON(g,t) == 1)

x(g+G\*(t-1))=0;

TON(g,t)=0;

if t==1

TOFF(g,t)=InitialTOFF(g)+1;

else

TOFF(g,t)=TOFF(g,t-1)+1;

end

else

continue;

end

else

break;

end

end

end

end