Cvx_project.m

```
load('data.mat')
ANPV=diag(Area)*NPV;
vht=diag(Area)*Vol;

tic
cvx_begin quiet
    variable x(N, T);
    variable VH(N, T);
    maximize(sum(sum(ANPV.*x)));
    %subject to
        x>=0; x<=1;
        VH==vht.*x;
        for j = 1:889
           sum(x(j, :)) == 1;
           for i=1:15
               if (ZerosVar(j,i)==1)
                   x(j, i)==0;
               end
           end
        end;
        for i = 2:T
           0.8*sum(VH(:,i-1))<= sum(VH(:,i));
           1.2*sum(VH(:,i-1))>= sum(VH(:,i));
        end;
cvx_end;
toc

%To make the comparisons easier
x_cvx=zeros(1,13335);
for i=1:889
    for j=1:15
       x_cvx(1, j+15*(i-1))=x(i,j);
    end
end

x_cvx=round(x_cvx', 4);
```

Linprog_project.m

```
load('data.mat')
ANPV=diag(Area)*NPV;
vht=diag(Area)*Vol;

tic
% f - Cost function
f=zeros(1,13335);
for i=1:889
    for j=1:15
       f(1, j+15*(i-1))=-ANPV(i,j);
    end
end

%A -> A*x<=B
```

```
A=zeros(28,13335);
for i=1:14
    for j=1:889
        % First inequality:
        A(i, 15*(j-1)+i)= 0.8*vht(j,i);
        A(i, 15*(j-1)+i+1)=- vht(j,i+1);
        % Second inequality:
        A(i+14, 15*(j-1)+i)= -1.2*vht(j,i);
        A(i+14, 15*(j-1)+i+1)= vht(j,i+1);
    end
end
B=zeros(1, 28);

%Aeq -> Aeq*x=Beq
Aeq=zeros(890,13335);
for i=1:889
    for j=1:15
        Aeq(i, 15*(i-1)+j)=1;
        if (ZerosVar(i,j)==1)
            Aeq(890, 15*(i-1)+j)=1;
        end
    end
end

%Beq -> Aeq*x=Beq
Beq=ones(1, 890);
Beq(890)=0;

x_lin=linprog(f, A, B, Aeq, Beq, zeros(13335, 1), ones(13335,1));

x_lin=round(x_lin, 4);
toc
```

Intlinprog_project.m

```
load('data.mat')
ANPV=diag(Area)*NPV;
vht=diag(Area)*Vol;
N=15;
T=15;
tic

% f - Cost function
f=zeros(1, N);
for i=1:N
    for j=1:T
       f(1, j+T*(i-1))=-ANPV(i+280,j);
    end
end

%A - A*x<=B
A=zeros(2*(T-1),N*T);
for i=1:T-1
    for j=1:N
        % First inequality:
```

```
        A(i, T*(j-1)+i)= 0.5*vht(j+280,i);
        A(i, T*(j-1)+i+1)=- vht(j+280,i+1);
        % Second inequality:
        A(i+14, T*(j-1)+i)= -1.5*vht(j+280,i);
        A(i+14, T*(j-1)+i+1)= vht(j+280,i+1);
    end
end
B=zeros(1, 2*(T-1));

%Aeq -> Aeq*x=Beq
Aeq=zeros(N+1,N*T);
for i=1:N
    for j=1:T
        Aeq(i, T*(i-1)+j)=1;
        if (ZerosVar(i+280,j)==1)
            Aeq(N+1, T*(i-1)+j)=1;
        end
    end
end

%Beq -> Aeq*x=Beq
Beq=ones(1, N+1);
Beq(N+1)=0;

x1=intlinprog(f, 1:N*T, A, B, Aeq, Beq, zeros(N*T, 1), ones(N*T,1));
toc
```

Analyze.m

```
%Computes "Pick the highest" and Monte carlos for both Linprog and CVX
%These functions need to have run before
result = x_cvx==x_lin;

fprintf('The CVX and linprog method match in %d of the 13335
values.\n\n', sum(result));

%Pick the highest %-------------------------------------------------------
--------
x_cvxr=x_cvx;
for j=1:889
    HV=zeros(1, 2);
    Stop=0;
    for i=1:15
        if (x_cvx(15*(j-1)+i, 1)==1)
            Stop=1;
            break;
        else
            x_cvxr(15*(j-1)+i, 1)=0;
            if(x_cvx(15*(j-1)+i, 1)>HV(1))
                HV(1)=x_cvx(15*(j-1)+i, 1);
                HV(2)=i;
            end
        end
    end
    if(~Stop)
```

```
        x_cvxr(15*(j-1)+HV(2), 1)=1;
    end
end

x_linr=x_lin;
for j=1:889
    HV=zeros(1, 2);
    Stop=0;
    for i=1:15
        if (x_lin(15*(j-1)+i, 1)==1)
            Stop=1;
            break;
        else
            x_linr(15*(j-1)+i, 1)=0;
            if(x_lin(15*(j-1)+i, 1)>HV(1))
                HV(1)=x_lin(15*(j-1)+i, 1);
                HV(2)=i;
            end
        end
    end
    if(~Stop)
        x_linr(15*(j-1)+HV(2), 1)=1;
    end
end


result = x_cvxr==x_linr;

fprintf('After "picking the highest", CVX and linprog methods match in %d
of the 13335 values.\n\n', sum(result));

%Monte Carlos %----------------------------------------------------------
------------------
x_cvx_m=x_cvx;
for j=1:889
    Mc=zeros(1, 15);
    Stop=0;
    for i=1:15
        if (x_cvx(15*(j-1)+i, 1)==1)
            Stop=1;
            break;
        else
            Mc(i)=x_cvx(15*(j-1)+i, 1)+sum(Mc(1:i));
        end
    end
    if(~Stop)
        R=rand;
        Looking=1;
        for f=1:15
            if(Mc(f)>R && Looking)
                x_cvx_m(15*(j-1)+f, 1)=1;
                Looking=0;
                continue;
            end
            x_cvx_m(15*(j-1)+f, 1)=0;
        end
    end
end

%Monte Carlos
x_lin_m=x_lin;
for j=1:889
    Mc=zeros(1, 15);
    Stop=0;
    for i=1:15
        if (x_lin(15*(j-1)+i, 1)==1)
            Stop=1;
            break;
        else
            Mc(i)=x_lin(15*(j-1)+i, 1)+sum(Mc(1:i));
        end
    end
    if(~Stop)
        R=rand;
        Looking=1;
        for f=1:15
            if(Mc(f)>R && Looking)
                x_lin_m(15*(j-1)+f, 1)=1;
                Looking=0;
                continue;
            end
            x_lin_m(15*(j-1)+f, 1)=0;
        end
    end
end

result = x_cvx_m==x_lin_m;
fprintf('After aplying Monte Carlos to both methods, CVX and Linprog
method match in %d of the 13335 values.\n', sum(result));
```

CVX_adj_project.m

```
load('data.mat')
ANPV=diag(Area)*NPV;
vht=diag(Area)*Vol;
adj_rows = size(Adj,1);

cvx_begin quiet
    variable x(N, T);
    variable VH(N, T);
    maximize(sum(sum(ANPV.*x)));
    %subject to
        x>=0; x<=1;
        VH==vht.*x;
        for j = 1:889
           sum(x(j, :)) == 1;
           for i=1:15
               if (ZerosVar(j,i)==1)
                   x(j, i)==0;
               end
           end
        end;
        for i = 2:T
           0.8*sum(VH(:,i-1))<= sum(VH(:,i));
           1.2*sum(VH(:,i-1))>= sum(VH(:,i));
        end;
        %adjacency constraint
        for t = 1:T
            for r = 1:adj_rows
                x(Adj(r,1),t) + x(Adj(r,2),t) <= 1;
            end;
        end;
        %end adjacency st

cvx_end;

x_cvx=zeros(1,13335);
for i=1:889
    for j=1:15
       x_cvx(1, j+15*(i-1))=x(i,j);
    end
end
```