Homework 3 - updated

Due 2023.12.8
ME7129 Optimization in Engineering,
National Taiwan University.

Problem 1 (60%)

Please find the Pareto optima of the following bi-objective design problems

$$
\begin{aligned}
\min\{f_1, f_2\} \\
f_1 = x_1 + x_2 \ , f_2 = -10x_1 + x_2 \\
\text{s. to.} \quad g_1 = \frac{x_1^2 x_2}{20} - 1 \le 0 \\
g_2 = \frac{(x_1 + x_2 - 5)^2}{30} - \frac{(x_1 - x_2 - 12)^2}{120} - 1 \le 0 \\
g_3 = \frac{80}{x_1^2 + 8x_2 + 5} - 1 \le 0 \\
0 \le \{x_1, x_2\} \le 10
\end{aligned}
\tag{1}
$$

Files: *HW3_objective1*、*HW3_objective2*、*HW3_constrain*、*f1_value_at_x2_star*、*f2_value_at_x1_star*、*hw3_1*

- *HW3_objective1*: objective function f1
- *HW3_objective2*: objective function f2
- *HW3_constrain*: nonlinear constrain {g1, g2, g3}
- *f1_value_at_x2_star*: finding the point $\{f_2^*, f_1(x_2^*)\}$
- *f2_value_at_x1_star*: finding the point $\{f_1^*, f_2(x_1^*)\}$
- *hw3_1*: main scrip

HW3_objective1 :

```
function f1=HW3_objective1(x)
f1=x(1)+x(2);
```

HW3_objective2 :

```
function f2=HW3_objective2(x)
f2=-10*x(1)+x(2);
```

HW3_constrain :

```
function [c,ceq]=HW3_constrain(x)
c(1)=-(x(1).^2*x(2)/20-1);
c(2)=-((x(1)+x(2)-5).^2/30+(x(1)-x(2)-12).^2/120-1);
c(3)=-(80/(x(1).^2+8*x(2)+5)-1);
ceq=[];
```

f1_value_at_x2_star :

```
function [f2_str,f1_x2_str,x2_str]=f1_value_at_x2_star()
fun=@HW3_objective2;
x0=[5;5];
A=[];
b=[];
Aeq=[];
beq=[];
lb=[0;0];
ub=[10;10];
nonlcon=@HW3_constrain;
[x2_str,f2_str]=fmincon(fun,x0,A,b,Aeq,beq,lb,ub,nonlcon);
f1_x2_str=HW3_objective1(x2_str);
```

f2_value_at_x1_star :

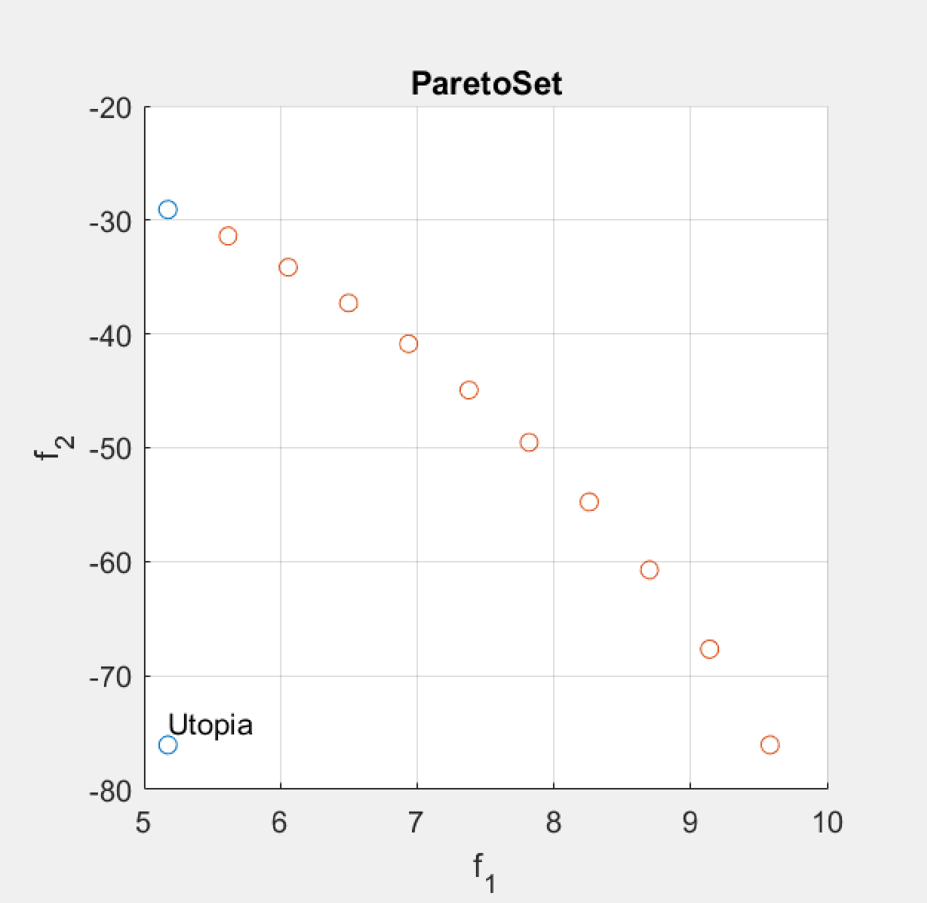
```
function [f1_str,f2_x1_str,x1_str]=f2_value_at_x1_star()
fun=@HW3_objective1;
x0=[5;5];
A=[];
b=[];
Aeq=[];
beq=[];
lb=[0;0];
ub=[10;10];
nonlcon=@HW3_constrain;
[x1_str,f1_str]=fmincon(fun,x0,A,b,Aeq,beq,lb,ub,nonlcon);
f2_x1_str=HW3_objective2(x1_str);
```

hw3_1 :

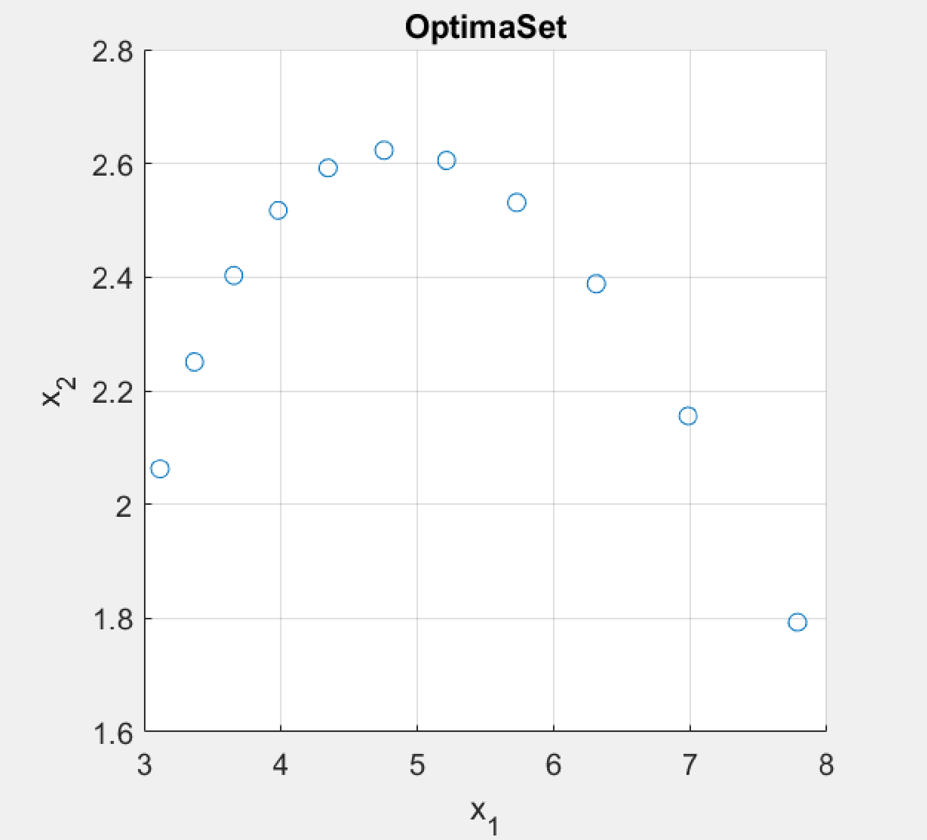
```
[f1_str,f2_x1_str,x1_str]=f2_value_at_x1_star(); % find {f^*_2 ,f1(x^*_2)}
[f2_str,f1_x2_str,x2_str]=f1_value_at_x2_star(); % find {f^*_1 ,f2(x^*_1)}
```

```
% plot three points
f1=[f1_str,f1_x2_str,f1_str];
f2=[f2_x1_str,f2_str,f2_str];
figure('position',[15,60,500,900])
subplot(2,1,1)
scatter(f1,f2);
axis square
hold on
text(f1_str,f2_str+2,"Utopia")

% devide (f^*_1) ~ (f2(x^*_1)) into N points, dosen't include (f^*_1)
N=10;
f1i=zeros(1,N);
for ii=1:N
    if ii==1
        f1i(1)=f1_str+(f1_x2_str-f1_str)/N;
    else
        f1i(ii)=f1i(ii-1)+(f1_x2_str-f1_str)/N;
    end
end


% using fmincon optimize algorithm find (f_2i)
f2i=zeros(1,N);
optima_x=zeros(2,N);
for ii=1:N
    fun=@HW3_objective2;
    x0=[5;5];
    A=[1,1];
    b=f1i(ii);
    Aeq=[];
    beq=[];
    lb=[0;0];
    ub=[10;10];
    nonlcon=@HW3_constrain;
    [optima_x(:,ii),f2i(ii)]=fmincon(fun,x0,A,b,Aeq,beq,lb,ub,nonlcon);
end
```

```
% plot Pareto set
scatter(f1i,f2i);
title ParetoSet
grid on
xlabel f_1
ylabel f_2
hold off

% plot the optima x at each Pareto set
subplot(2,1,2)
scatter([x1_str(1) optima_x(1,:)],[x1_str(2) optima_x(2,:)])
axis square
title OptimaSet
grid on
xlabel x_1
ylabel x_2
```



ParetoSet
-20
-30
-40
-50
-60
-70
-80
f_2
Utopia
5
6
7
8
9
10
f_1



OptimaSet
2.8
2.6
2.4
2.2
2
1.8
1.6
x_2
3
4
5
6
7
8
x_1

Problem 2 (40%)

Consider the problem

$$\begin{aligned}\min f &= x_1 + x_2\\ \text{s. to.}\quad g_1 &= 1 - \frac{x_1^2 x_2}{20} \le 0\\ g_2 = 1 - \frac{(x_1+x_2-5)^2}{30} &- \frac{(x_1-x_2-12)^2}{120} \le 0\\ g_3 &= 1 - \frac{80}{x_1^2 + 8x_2 + 5} \le 0\\ 0 \le x_1, &x_2 \le 10\end{aligned} \tag{2}$$

Assume that the final optimal might have manufacturing uncertainties with $X \sim N(\mu_x, \sigma_x^2)$ where $[\mu_{X_1}, \mu_{X_2}] = [x_1^*, x_2^*]$, and $\sigma_{X_1} = \sigma_{X_2} = 0.3$.

1. Please run Monte Carlo simulations with 10 samples, what are the probability values of the optimal violating each constraints? 15%
2. Repeat previous 10-run MCS for 20 times, do you get the same results every time? Why not? 15%
3. Please run Monte Carlo simulations with 1 million samples, what are the probability values of the optimal violating each constraints? 5%
4. Repeat previous 1 million-run MCS for 20 times, do you get the same results every time? Why or why not? 10 %
5. Please explain the difference between 10 sample MCS and 1 million sample MCS. 15%

Files: *HW3_optimization*、*hw3_2_1*、*hw3_2_2*、*hw3_2_3*、*hw3_2_4*

HW3_optimization：

```
function x=HW3_optimization()
fun=@HW3_objective1;
x0=[5;5];
A=[];
b=[];
Aeq=[];
beq=[];
lb=[0;0];
ub=[10;10];
nonlcon=@HW3_constrain;
x=fmincon(fun,x0,A,b,Aeq,beq,lb,ub,nonlcon);
```

2-1.

Hw3_2_1:

```
optimal_design=HW3_optimization();
mux=optimal_design'; % you should change to the optimal design you obtained
stdx=[0.3, 0.3]; % you should change this value according to the homework
descriptions
covX=[stdx(1)^2, 0; 0, stdx(2)^2];

% Basic MCS

N=10; % you should change this value according to the homework descriptions

RandX=mvnrnd(mux, covX, N);
X1=RandX(:,1);
X2=RandX(:,2);

%Y=zeros(N,1); % you should change Y to constraints, therefore you have
three different function evaluations
%Y=5*X1-3*X2;
g1=1-X1.^2.*X2/20;
g2=1-(X1+X2-5).^2/30-(X1-X2-12).^2/120;
g3=1-80/(X1.^2+8.*X2+5);

Nf(1)=sum(g1<0); % you should have three Nf values w.r.t different
constraints
Nf(2)=sum(g2<0);
Nf(3)=sum(g3<0);
pf=Nf./N; % you should have three pf values w.r.t different constraints

for ii=1:3
    sprintf('g%d Failure probability using MCS with %d samples is %0.5g
percent ', ii , N, pf(ii)*100)
end
```

result:

Command Window

```
ans =

    'g1 Failure probability using MCS with 10 samples is 60 percent '


ans =

    'g2 Failure probability using MCS with 10 samples is 50 percent '


ans =

    'g3 Failure probability using MCS with 10 samples is 10 percent '

fx >>
```

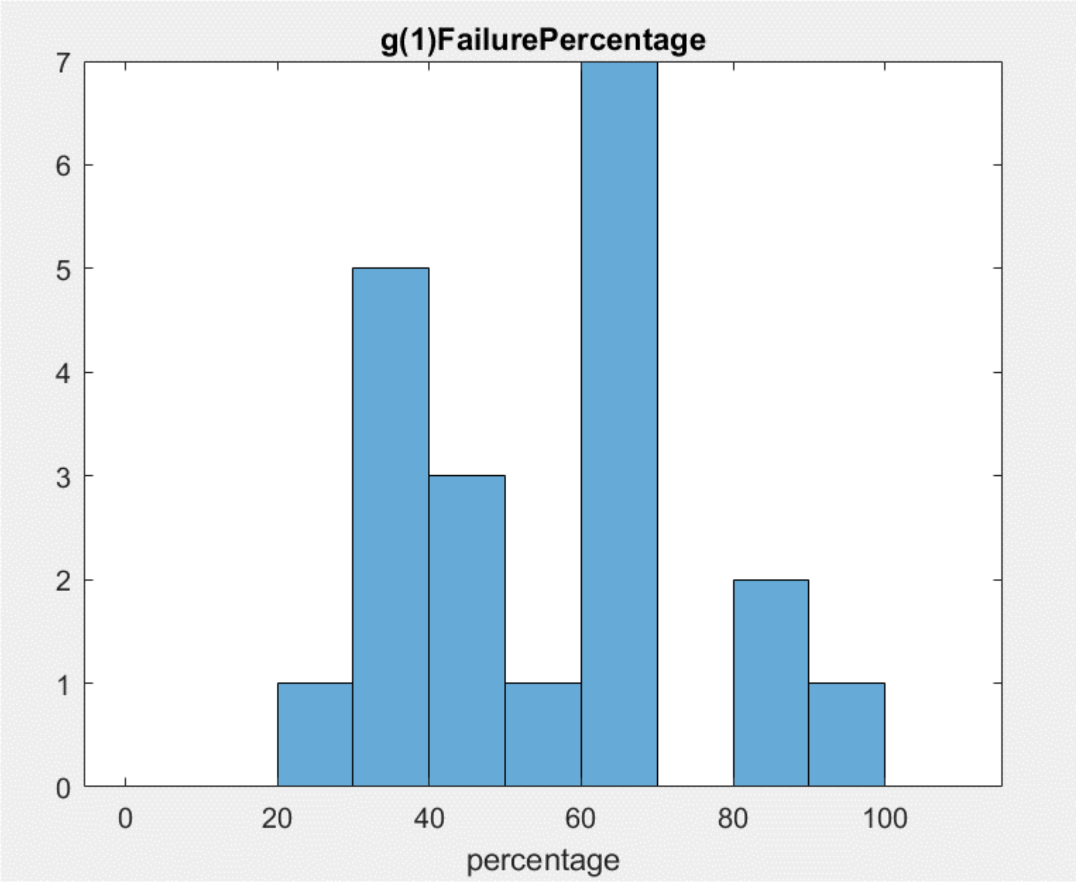
2-2.

Hw3_2_2 :

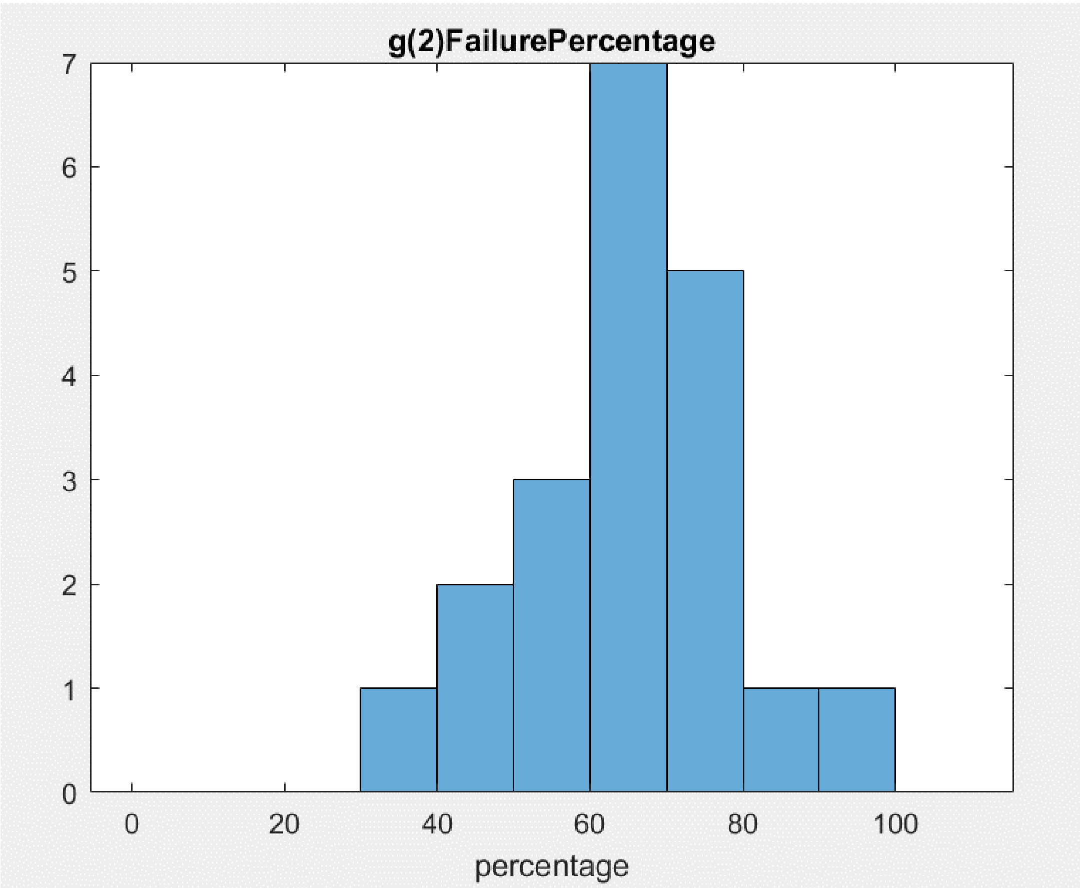
```
optimal_design=HW3_optimization();
mux=optimal_design'; % you should change to the optimal design you obtained
stdx=[0.3, 0.3]; % you should change this value according to the homework
descriptions
covX=[stdx(1)^2, 0; 0, stdx(2)^2];

% Basic MCS

N=10; % you should change this value according to the homework descriptions

percents=zeros(3,20);

for jj=1:20
    RandX=mvnrnd(mux, covX, N);
    X1=RandX(:,1);
    X2=RandX(:,2);

%Y=zeros(N,1); % you should change Y to constraints, therefore you have
three different function evaluations
%Y=5*X1-3*X2;
    g1=1-X1.^2.*X2/20;
    g2=1-(X1+X2-5).^2/30-(X1-X2-12).^2/120;
```

```
    g3=1-80/(X1.^2+8.*X2+5);

    Nf(1)=sum(g1<0); % you should have three Nf values w.r.t different
constraints
    Nf(2)=sum(g2<0);
    Nf(3)=sum(g3<0);
    pf=Nf./N; % you should have three pf values w.r.t different constraints

%     for ii=1:3
%         sprintf('Failure probability using MCS with %d samples is %0.5g
percent ', N, pf(ii)*100)
%     end

    percents(:,jj)=pf';
end

figure(1)
histogram(percents(1,:)*100,0:10:110)
title g(1)FailurePercentage
xlabel percentage
figure(2)
histogram(percents(2,:)*100,0:10:110)
title g(2)FailurePercentage
xlabel percentage
figure(3)
histogram(percents(3,:)*100,0:10:110)
title g(3)FailurePercentage
xlabel percentage

RE=string(percents*100);
SULT=repmat("%",3,20);
RESULT=RE+SULT;
sprintf("the failure percentage of each constrains for 20 times:")
disp(RESULT)
```

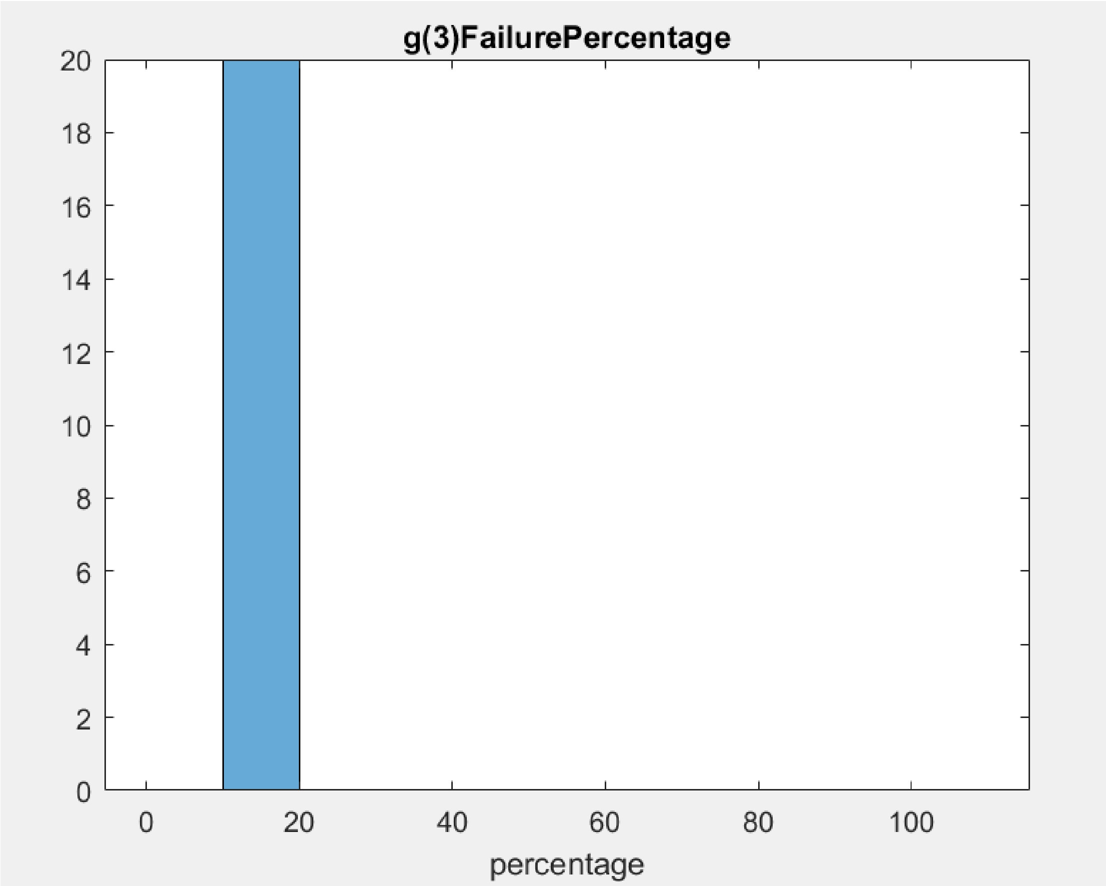
result:



g(1)FailurePercentage
7
6
5
4
3
2
1
0
0
20
40
60
80
100
percentage



g(2)FailurePercentage
7
6
5
4
3
2
1
0
0
20
40
60
80
100
percentage



Command Window

```
  "the failure percentage of each constrains for 20 times:"

Columns 1 through 14

  "50%"    "60%"    "60%"    "40%"    "30%"    "30%"    "60%"    "40%"    "60%"    "60%"    "60%"    "20%"    "80%"    "80%"
  "60%"    "40%"    "30%"    "60%"    "60%"    "70%"    "70%"    "70%"    "70%"    "50%"    "50%"    "40%"    "90%"    "60%"
  "10%"    "10%"    "10%"    "10%"    "10%"    "10%"    "10%"    "10%"    "10%"    "10%"    "10%"    "10%"    "10%"    "10%"

Columns 15 through 20

  "40%"    "30%"    "30%"    "30%"    "90%"    "60%"
  "60%"    "70%"    "60%"    "80%"    "50%"    "60%"
  "10%"    "10%"    "10%"    "10%"    "10%"    "10%"
```

I did not get the same results every time. I think it is because that the quantity of samples is too small. 10 samples each simulation can’t completely demonstrate the probability of failure.

2-3.

Hw3_2_3 :

```
optimal_design=HW3_optimization();
mux=optimal_design'; % you should change to the optimal design you obtained
stdx=[0.3, 0.3]; % you should change this value according to the homework descriptions
covX=[stdx(1)^2, 0; 0, stdx(2)^2];
```

```
% Basic MCS

N=10^6; % you should change this value according to the homework
descriptions

RandX=mvnrnd(mux, covX, N);
X1=RandX(:,1);
X2=RandX(:,2);

%Y=zeros(N,1); % you should change Y to constraints, therefore you have
three different function evaluations
%Y=5*X1-3*X2;
g1=1-X1.^2.*X2/20;
g2=1-(X1+X2-5).^2/30-(X1-X2-12).^2/120;
g3=1-80/(X1.^2+8.*X2+5);

Nf(1)=sum(g1<0); % you should have three Nf values w.r.t different
constraints
Nf(2)=sum(g2<0);
Nf(3)=sum(g3<0);
pf=Nf./N; % you should have three pf values w.r.t different constraints

for ii=1:3
    sprintf('g%d Failure probability using MCS with %d samples is %0.5g
percent ', ii , N, pf(ii)*100)
end
```

result:

Command Window

```
ans =

    'g1 Failure probability using MCS with 1000000 samples is 48.351 percent '


ans =

    'g2 Failure probability using MCS with 1000000 samples is 53.099 percent '


ans =

    'g3 Failure probability using MCS with 1000000 samples is 0.0001 percent '
```

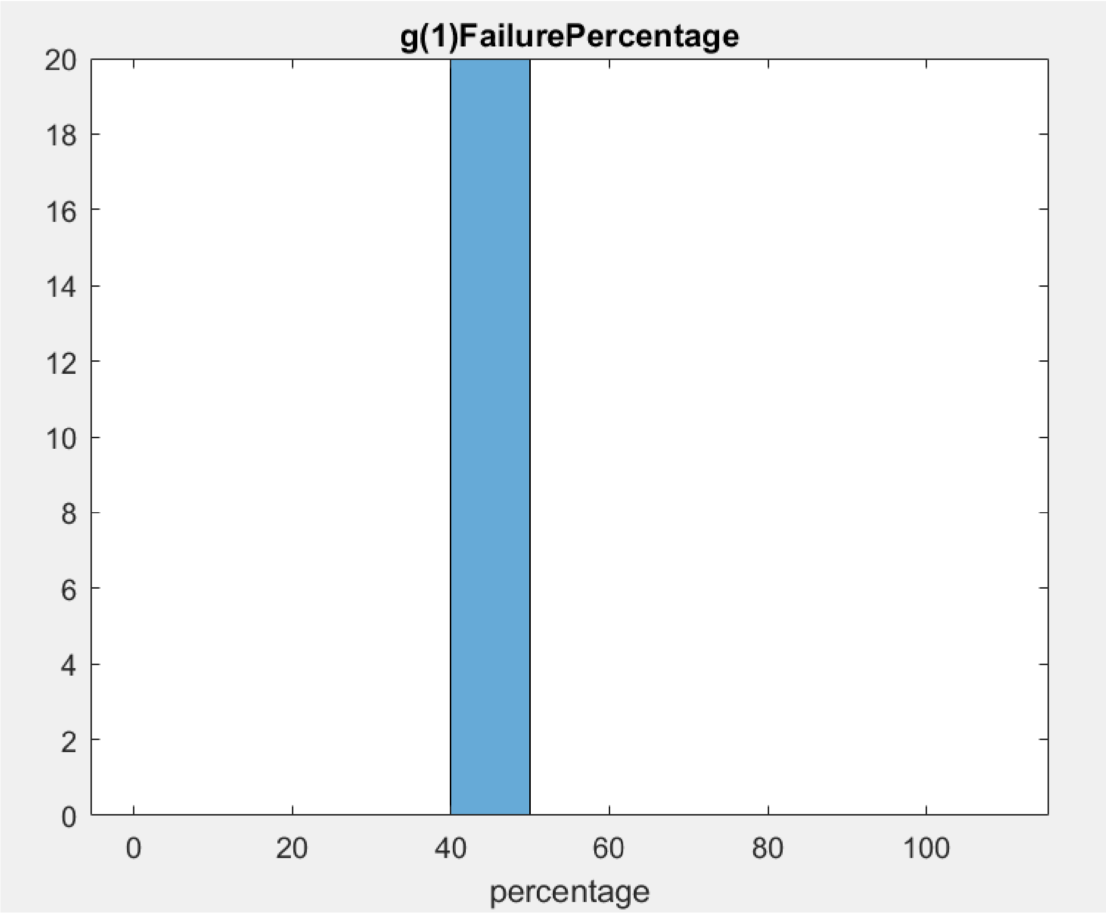
2-4.

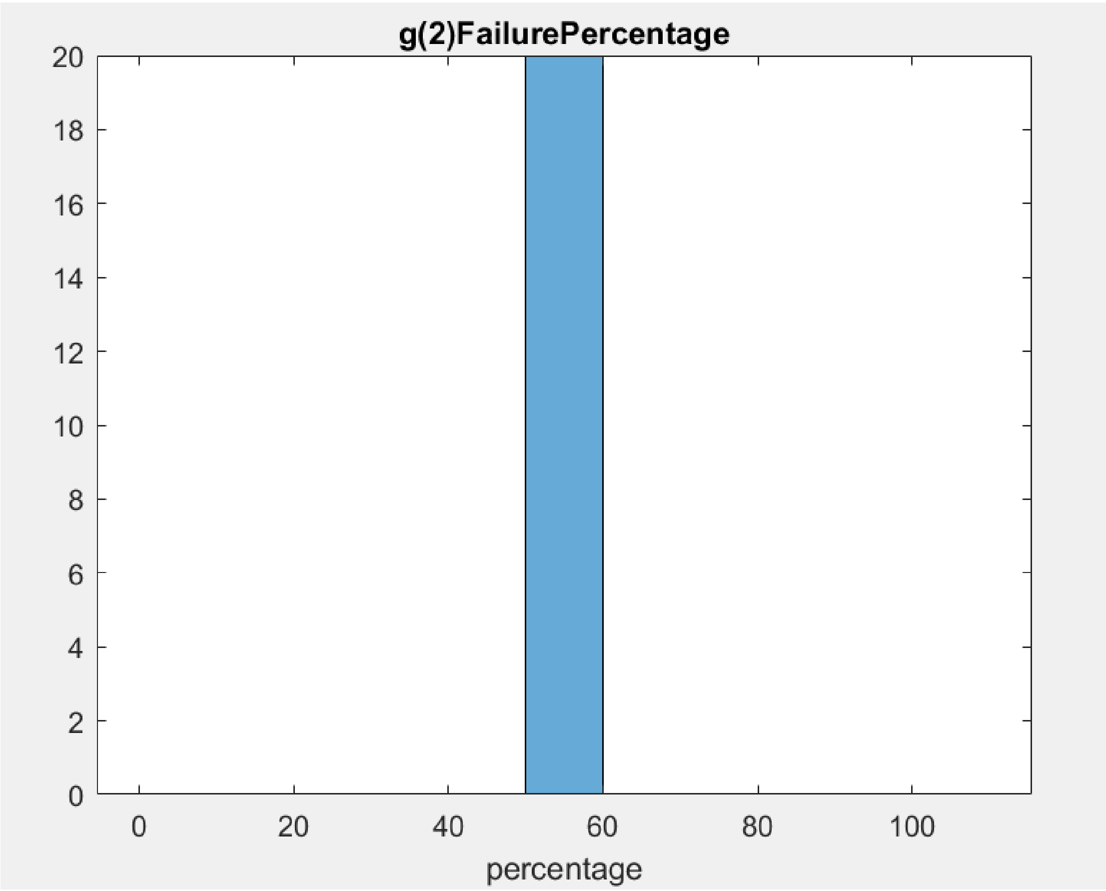
hw3_2_4:

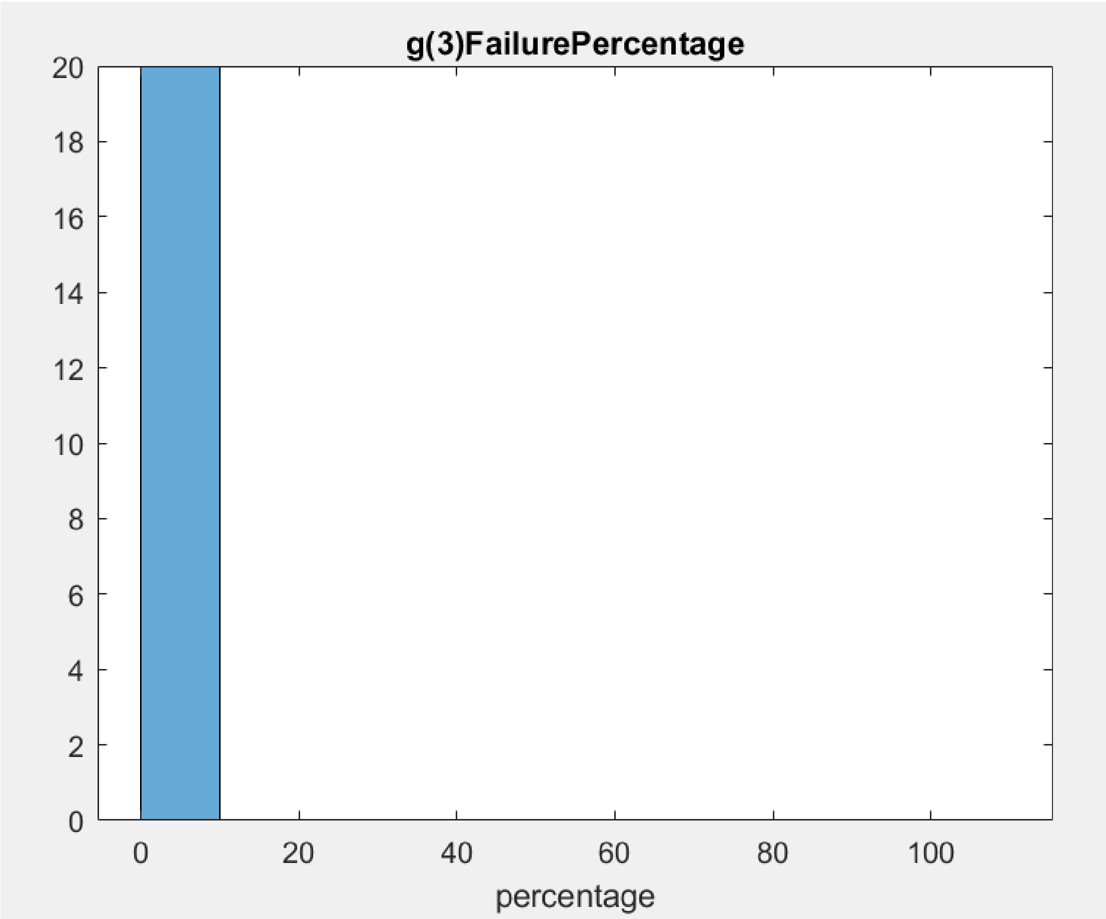
```
optimal_design=HW3_optimization();
mux=optimal_design'; % you should change to the optimal design you obtained
stdx=[0.3, 0.3]; % you should change this value according to the homework descriptions
covX=[stdx(1)^2, 0; 0, stdx(2)^2];

% Basic MCS

N=10^6; % you should change this value according to the homework descriptions

percents=zeros(3,20);

for jj=1:20
    RandX=mvnrnd(mux, covX, N);
    X1=RandX(:,1);
    X2=RandX(:,2);

%Y=zeros(N,1); % you should change Y to constraints, therefore you have three different function evaluations
%Y=5*X1-3*X2;
    g1=1-X1.^2.*X2/20;
    g2=1-(X1+X2-5).^2/30-(X1-X2-12).^2/120;
    g3=1-80/(X1.^2+8.*X2+5);

    Nf(1)=sum(g1<0); % you should have three Nf values w.r.t different constraints
    Nf(2)=sum(g2<0);
    Nf(3)=sum(g3<0);
    pf=Nf./N; % you should have three pf values w.r.t different constraints

%     for ii=1:3
%         sprintf('Failure probability using MCS with %d samples is %0.5g percent ', N, pf(ii)*100)
%     end
```

```
    percents(:,jj)=pf';
end

figure(1)
histogram(percents(1,:)*100,0:10:110)
title g(1)FailurePercentage
xlabel percentage
figure(2)
histogram(percents(2,:)*100,0:10:110)
title g(2)FailurePercentage
xlabel percentage
figure(3)
histogram(percents(3,:)*100,0:10:110)
title g(3)FailurePercentage
xlabel percentage

RE=string(percents*100);
SULT=repmat("%",3,20);
RESULT=RE+SULT;
sprintf("the failure percentage of each constrains for 20 times:")
disp(RESULT)
```

result:


g(1)FailurePercentage
20
18
16
14
12
10
8
6
4
2
0
0
20
40
60
80
100
percentage


g(2)FailurePercentage
20
18
16
14
12
10
8
6
4
2
0
0
20
40
60
80
100
percentage



Command Window

```
  "the failure percentage of each constrains for 20 times:"

Columns 1 through 9

  "48.3183%"    "48.3384%"    "48.3448%"    "48.2705%"    "48.265%"     "48.3284%"    "48.4325%"    "48.3504%"    "48.36%"
  "53.0981%"    "53.0884%"    "53.0901%"    "53.0234%"    "53.0271%"    "53.1855%"    "53.0883%"    "52.9931%"    "52.9904%"
  "0.0001%"     "0.0001%"     "0.0001%"     "0.0001%"     "0.0001%"     "0.0001%"     "0.0001%"     "0.0001%"     "0.0001%"

Columns 10 through 18

  "48.3534%"    "48.3216%"    "48.3839%"    "48.4142%"    "48.3673%"    "48.3871%"    "48.2685%"    "48.2975%"    "48.4109%"
  "53.0687%"    "53.09%"      "53.0248%"    "53.0249%"    "53.0982%"    "53.1002%"    "53.0313%"    "53.0733%"    "52.9559%"
  "0.0001%"     "0.0001%"     "0.0001%"     "0.0001%"     "0.0001%"     "0.0001%"     "0.0001%"     "0.0001%"     "0.0001%"

Columns 19 through 20

  "48.3346%"    "48.4314%"
  "53.106%"     "53.0204%"
  "0.0001%"     "0.0001%"
```

I almost got the same results every time. It is because that the quantity of samples is adequate to simulate the real probability of failure.

2-5.

The biggest difference between 10 sample MCS and 1 million sample MCS are the stability of results. 10 sample MCS results to different failure percentages each run, while 1 million sample MCS has almost same result each run.