

**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

*f2\_value\_at\_x1\_star*: finding the point

*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,:)])

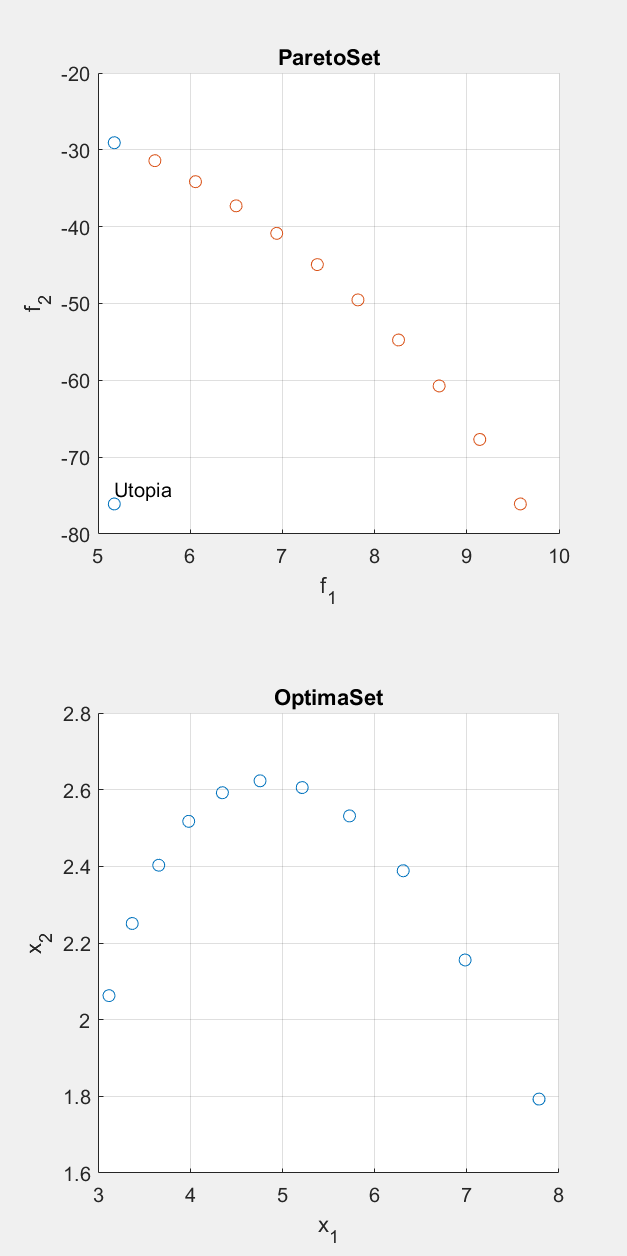
axis square

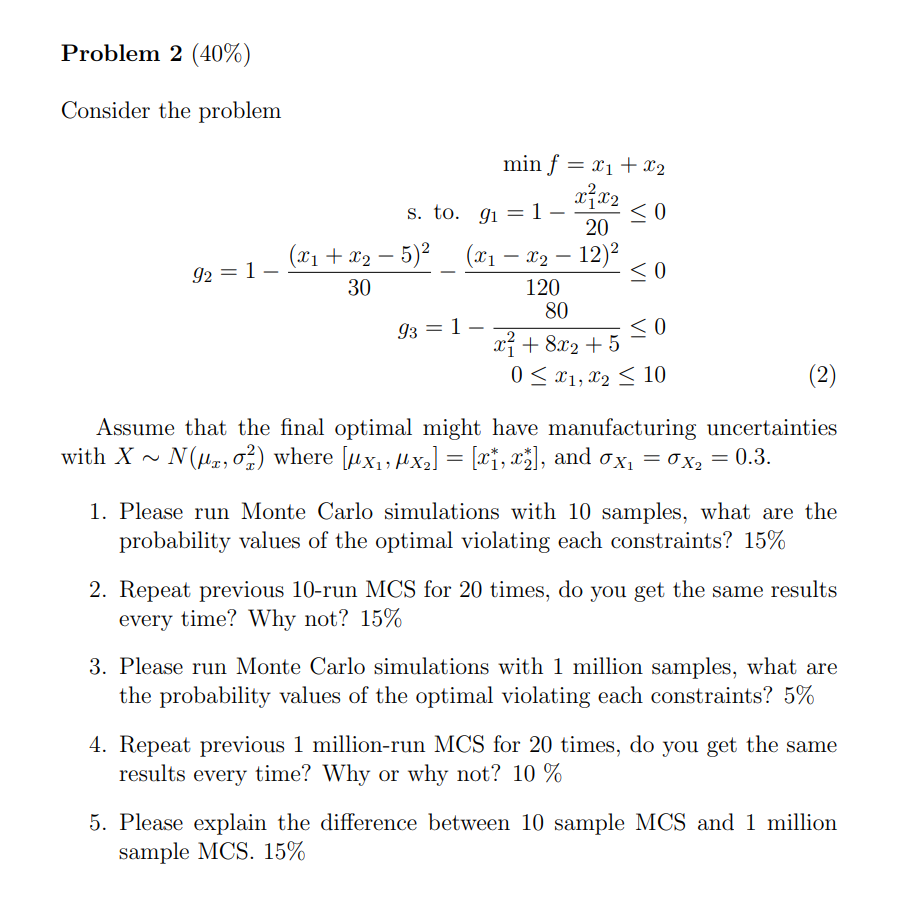
title OptimaSet

grid on

xlabel x\_1

ylabel x\_2





**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);

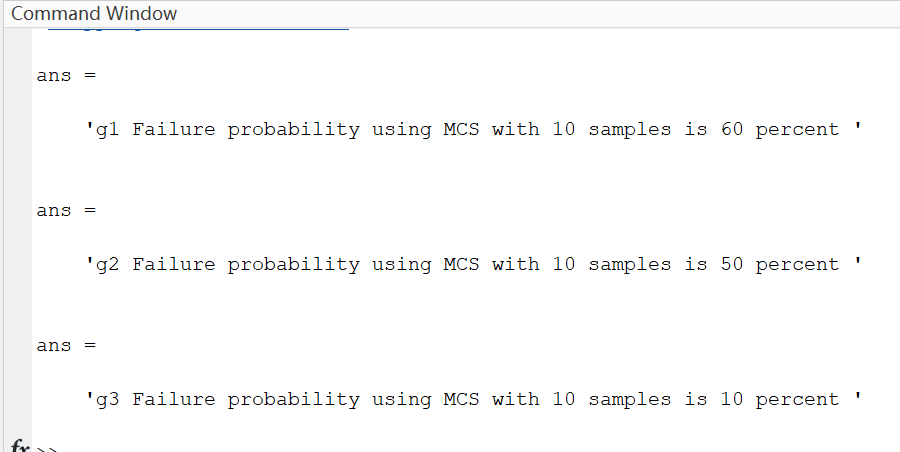
pf=Nf./N; % you should have three pf values w.r.t different constraints

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:**



**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);

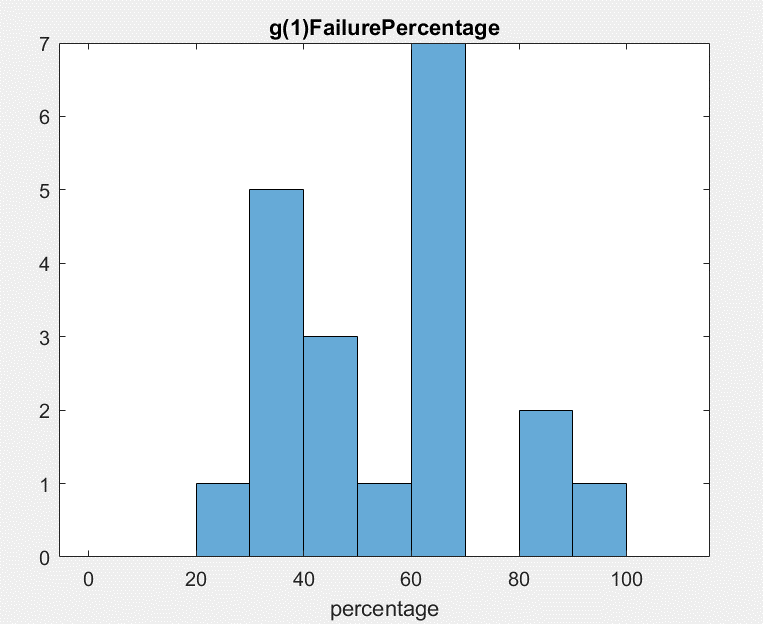
SULT=repmat("%",3,20);

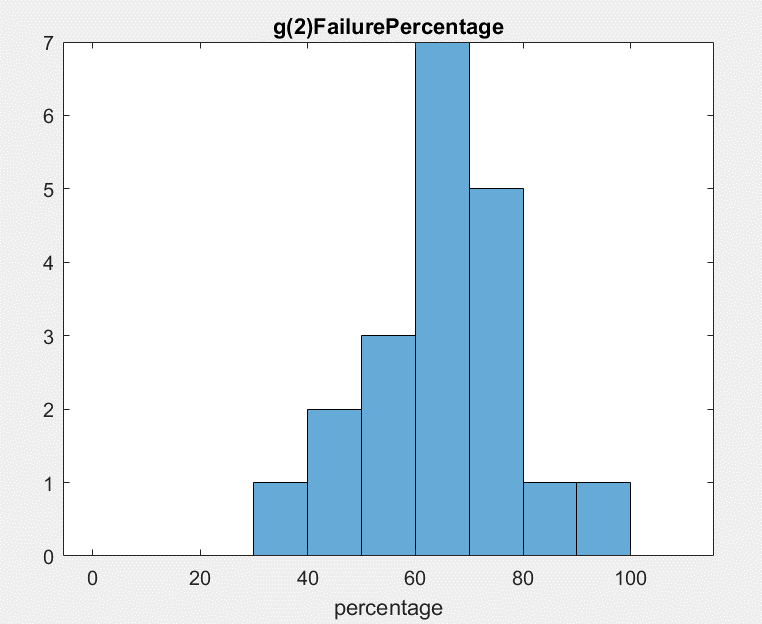
RESULT=RE+SULT;

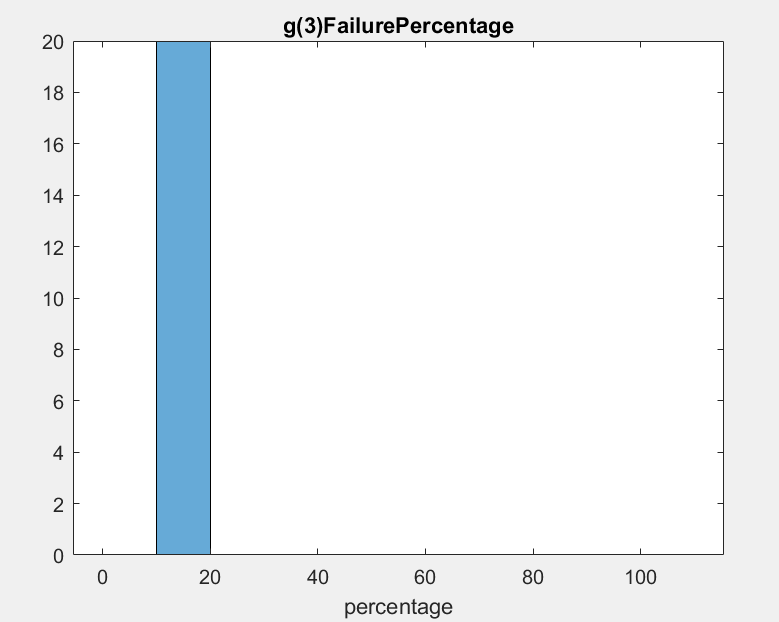
sprintf("the failure percentage of each constrains for 20 times:")

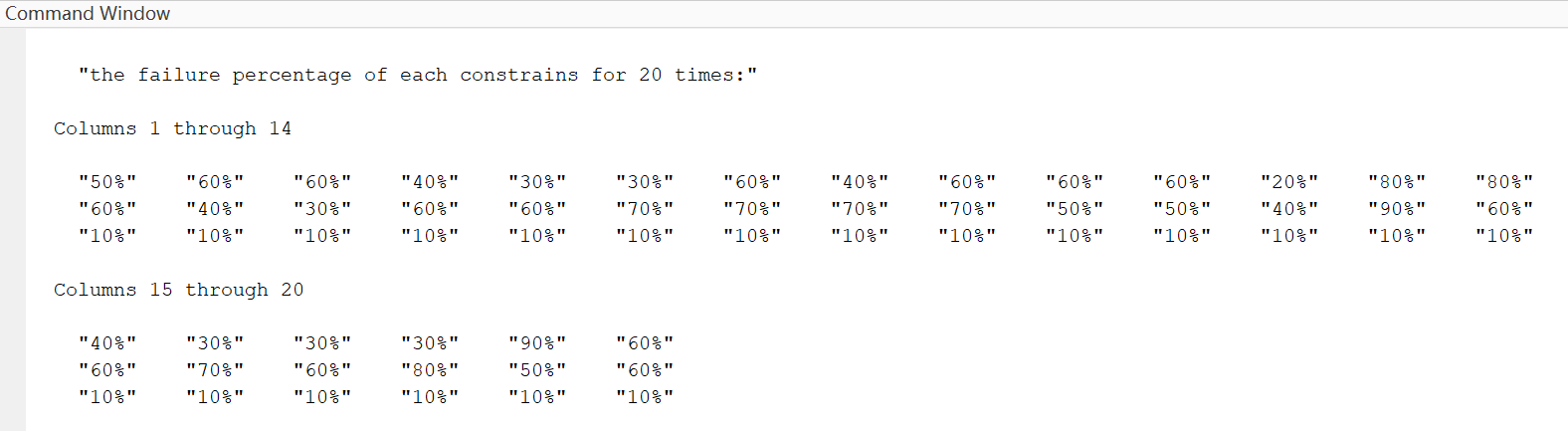
disp(RESULT)

**result:**









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);

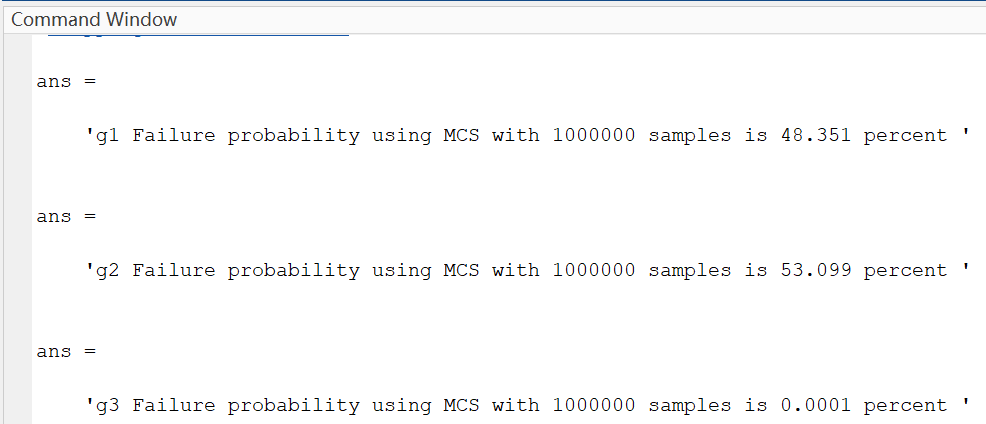
pf=Nf./N; % you should have three pf values w.r.t different constraints

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:**



**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);

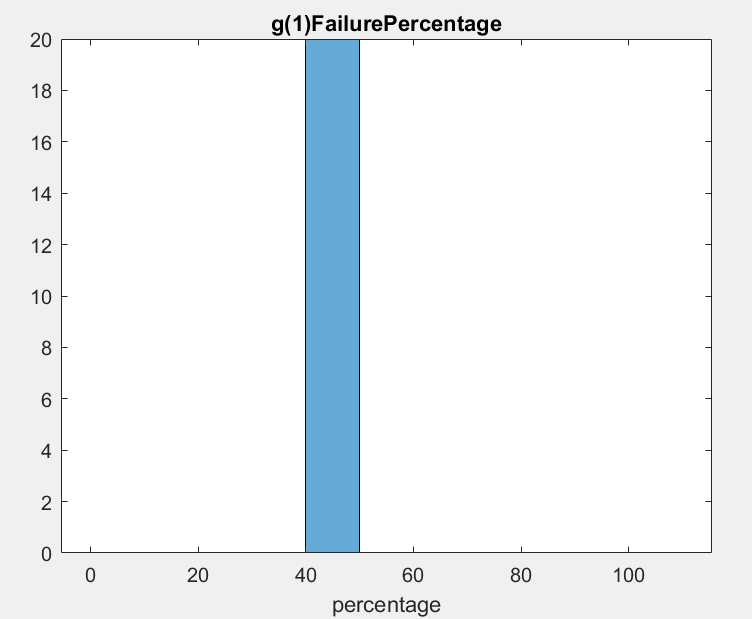
SULT=repmat("%",3,20);

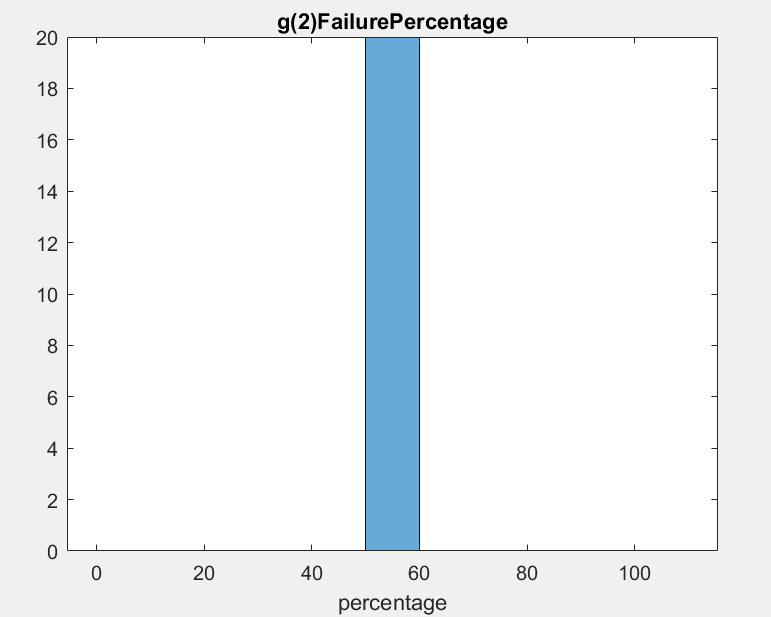
RESULT=RE+SULT;

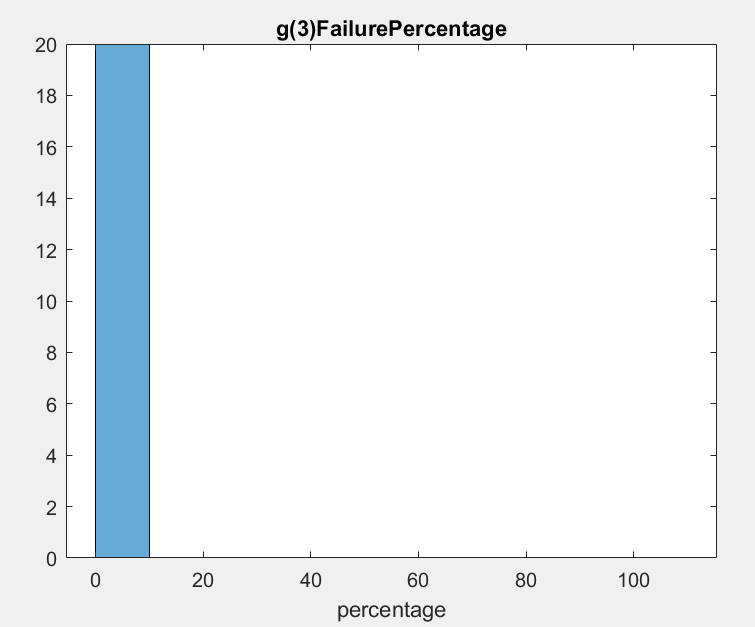
sprintf("the failure percentage of each constrains for 20 times:")

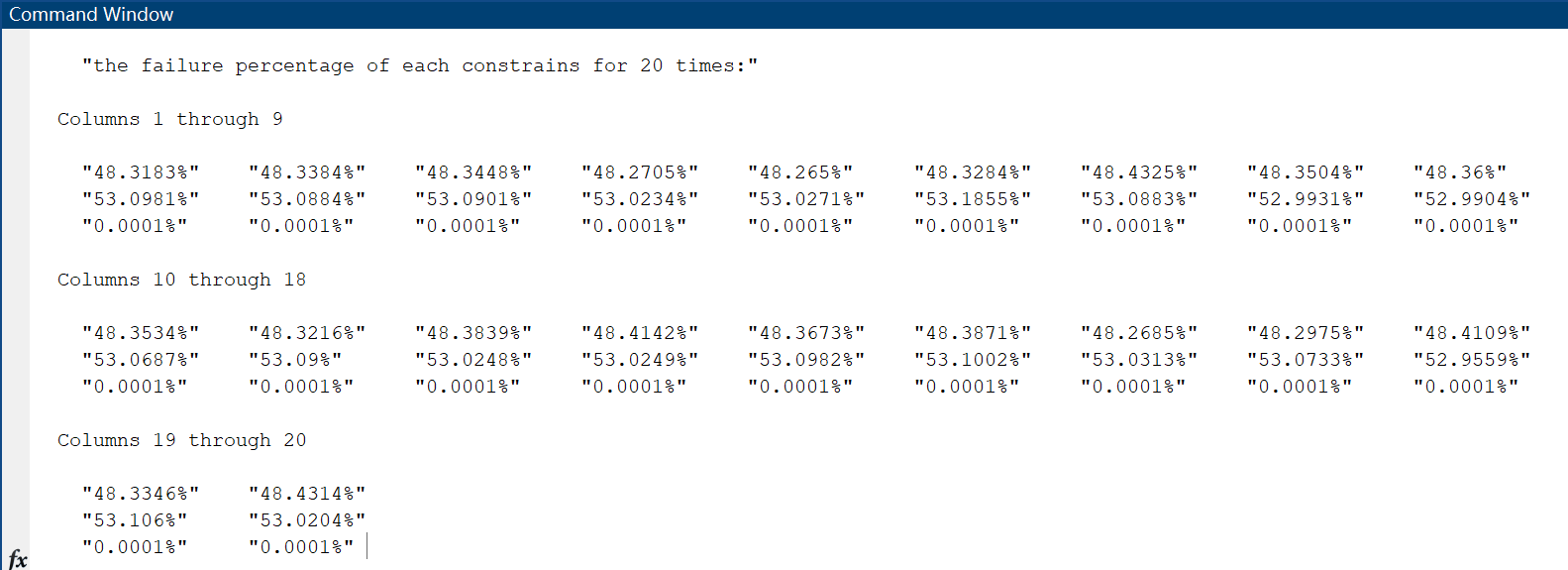
disp(RESULT)

**result:**









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.