Starting from an existing code, the objective is to implement a new selection principle, called stochastic ranking in order to handle constraints in a different way.

I. The initial script .

The initial script is given in annex. It is a real valued genetic algorithm, with no elitism, aimed to solve the ‘can problem’ with a fixed-penalty method (that is design a cylindrical can having a minimal surface with a volume greater than 300ml).

1. Explain the definition of the cost function in the script.
2. Describe and justify the mutation operator. In what sense can we say it is an adaptative mutation?
3. Describe the crossover operator. What variant of this crossover could you propose?
4. Comment the two figures that are plotted after the execution of this script.

II. The stochastic ranking .

Stochastic ranking is a method that is described below to rank \( \lambda \) individuals in order to take into account the constraints in the ranking but with some randomness.

In this algorithm, \( f \) is the cost function that needs to be minimized (for instance in the can problem, the surface) and \( \Phi \) is the penalty term (for instance, \( \min(V - 300, 0)^2 \) for the can problem). In this algorithm, the fixed parameter \( P_f \) is responsible for the randomness in the ranking.

1. Describe the stochastic ranking when \( P_f = 0 \), respectively 1.
2. Write in a Scilab language the stochastic ranking of \( \lambda \) elements with a given function \( f \) and a penalty function \( \Phi \).
IV The Scilab script:

function Xeval=evaluation(X);
mu=1;
[Npop,n2]=size(X);
Xeval=X;
for i=1:Npop;
    penal=min(%pi*X(i,1)^2*X(i,2)/4-300,0)^2;
    Xeval(i,$)=%pi*X(i,1)^2/4+%pi*X(i,1)*X(i,2)+mu*penal;
end
endfunction

function bestX=best(X);
[Npop,n2]=size(X);
[y,k]=gsort(-X(:,$));
bestX=X(k(1),:);
endfunction

function Xsel=selection(X);
[Npop,n2]=size(X);
[y,k]=gsort(X(:,$))
X=X(k,:);
p=1:Npop;
roulette=cumsum(p)/sum(p);
Xsel=[];
for i=1:Npop
  u=rand();isel=1;
  while (u>roulette(isel))
    isel=isel+1;
  end
  Xsel=[Xsel;X(isel,:)];
end
endfunction

function xcrois=croisement(x)
xcrois=x;
u=rand();
xcrois(1,:)=u*x(1,:)+(1-u)*x(2,:);
xcrois(2,:)=(1-u)*x(1,:)+u*x(2,:);
endfunction

function xmut=mutation(x,gen,Ngen,xmin,xmax)
bet=5;
if (rand()<1/2) then
  xmut=x+rand()*(xmax-x)*(1-(gen-1)/Ngen)^bet;
else
  xmut=x-rand()*(x-xmin)*(1-(gen-1)/Ngen)^bet;
end
endfunction

/////////MAIN /////////////////////////////////////////////////////////////
Npop=300;Ngen=40;
n=2;
pc=0.2;pm=0.3;
xmin=1;xmax=20;
X=xmin+(xmax-xmin)*rand(Npop,n+1);
bestX=[];
for gen=1:Ngen
  X=evaluation(X); // evaluation
  X=selection(X); // selection
  bestX=[bestX;best(X)];
end
Xnew=[];
for j=1:Npop/2
u1=int(Npop*rand())+1;
u2=int(Npop*rand())+1;
if (rand()<pc) then
    Xnew=[Xnew;croisement(X([u1,u2],:))]; //croisement
else
    Xnew=[Xnew;X([u1,u2],:)];
end
end
for j=1:Npop
    if (rand())<pm) then
        Xnew(j,:)=mutation(Xnew(j,:),gen,Ngen,xmin,xmax); // mutation
    end
end
X=Xnew;
edn
//////////TRACE //////////////////////////
function area = can(x)
d=x(1);h=x(2);
area=%pi*d^2/4+%pi*d*h
endfunction
///////////////////////////////
xset('window',0);clf
plot2d(Npop*(1:Ngen),bestX(:,$))

/////////////////////////////////////////
xset('window',1);clf;
xmin=1;xmax=12;ymin=1;ymax=12;
xcan=xmin:((xmax-xmin)/100):xmax;
ycan=ymin:((ymax-ymin)/100):ymax;
for i=1:101;
    for j=1:101;
        zcan(i,j)=can([xcan(i),ycan(j)]);
    end
    hcan(i)=300*4/(%pi*xcan(i)^2);
end
plot2d(bestX(:,1),bestX(:,2),rect=[xmin,ymin,xmax,ymax]);
plot2d(bestX($,1),bestX($,2),-2);
plot2d(xcan,hcan,2,rect=[xmin,ymin,xmax,ymax]);
contour2d(xcan,ycan,zcan,10,rect=[xmin,ymin,xmax,ymax]);