Multiobj – Linear function Linear constraint (funzione esempio)

```
% min Cx % Ax <= b
C = [ 1  -1   %f1(x)
      1  1] ; %f2(x)
A =[ -2    1 %Ax <=b
   -1     -1
    5    -1];
b = [ 0
      0
      6] ;
% % solve the scalarized problem with 0 < alfa < 1
MINIMA=[];
LAMBDA=[];
for alfa = 0.01 : 0.01 : 0.99
    [x,fval,exitflag,output,lambda] = linprog(alfa*C(1,:)+(1-alfa)*C(2,:),A,b) ;
    MINIMA=[MINIMA; x'];
    LAMBDA=[LAMBDA;alfa,lambda.ineqlin'];
end
%le x(i) le metto per quante ci sono
fprintf('\t x(1) \t x(2) \t alpha  \t  LAMBDA \n\n'); %le X le modifico in base a quante ci sono
       [MINIMA , LAMBDA]
% % solve the scalarized problem with alfa = 0
alfa = 0;
[xalfa0,f0,exitflag,output,lambda0] = linprog(alfa*C(1,:)+(1-alfa)*C(2,:),A,b);
xalfa0
% % solve the scalarized problem with alfa = 1
alfa = 1;
[xalfa1,f1,exitflag,output,lambda1] = linprog(alfa*C(1,:)+(1-alfa)*C(2,:),A,b);
xalfa1
```

NON Linear function Linear constraint (funzione esempio)

```
clear all;
Q1 = [2 0; 0 2] ; Q2 = [2 0; 0 2] ;
c1=[2 -4]';
c2=[-6 -4]';
A =[ 0 -1; -2 1; 2 1 ]; b = [0 0 4]';
% solve the scalarized problem with  alfa1 in [0,1]
MINIMA=[ ]; % First column: value of alfa1
LAMBDA=[ ]; % First column: value of alfa1
for alfa1 = 0.01 : 0.001 : 1
[x,fval,exitflag,output,lambda] = quadprog(alfa1*Q1+(1-alfa1)*Q2,alfa1*c1+(1-alfa1)*c2,A,b) ;
MINIMA=[MINIMA; alfa1 x'];
LAMBDA=[LAMBDA;alfa1,lambda.ineqlin'];
end
 %mi da disegno che raffigura pareto min (devo vedere se weak o no)
 plot(MINIMA(:,2),MINIMA(:,3), 'r*')
 fprintf('\t alpha \t x(1) \t x(2) \t alpha  \t  LAMBDA \n\n');
                         [MINIMA , LAMBDA]
```

nonlinear constraints (funzione esempio)

```
MINIMA=[ ]; % First column: value of alfa1  second and then x(i)
LAMBDA=[ ];
% solve the scalarized problem with 0 =< alfa1 <= 1
for alfa1 = 0 : 0.01 : 1
   FUN=@(x) (2*alfa1-1)*x(1)+x(2); %funzione scalarizzata
   NONLINCON= @(x) const(x);
   [x,fval,exitflag,output,lambda] = fmincon(FUN,[0;0],[],[],[],[],[],[],NONLINCON) ;
   MINIMA=[MINIMA; alfa1, x'];
    LAMBDA=[LAMBDA; alfa1, lambda.ineqnonlin’];
end
plot(MINIMA(:,2),MINIMA(:,3)) %disegna pareto border
 fprintf('\t alpha \t x(1) \t x(2) \t alpha  \t  LAMBDA \n\n');
      [MINIMA , LAMBDA]
function [C,Ceq]=const(x)
C=[x(1)^2 +x(2)^2 –1];  %constraint vanno qui
Ceq=[];
end
```

Goal method (funzione esempio)

```
clear all
C = [1 -1; 1 1] ;
A =[ -2 1; -1 -1; 5 -1 ];
b = [0 0 6]';
% find the ideal point z
z=[0,0]';
for i = 1:2
   [a,z(i)]=linprog(C(i,:)',A,b);
end
fprintf('Ideal point')
 z
% solve the quadratic problem with norm q=2
x=quadprog(C'*C,-C'*z,A,b) %optimal solution
```