ADMM (Alternating Direction Method of Multipliers)

Let us consider the following opt-prob.

Augmented Lagrangian:

|  |
| --- |
| THE ADMM ALGO: (Assume: f and g are cvx fcns) |
|  |

Example:

|  |  |
| --- | --- |
| %% ADMM problem  clear all,close all,clc;yalmip('clear');  % REQUIRED TOOLS: YALMIP,MOSEK  %% problem construction  x\_real=[1:1:10]'; x\_dim=length(x\_real);  z\_real=[1:1:20]'; z\_dim=length(z\_real);  rng(123)  Qx=randi([-10,10],x\_dim,x\_dim);  Qx=Qx'\*Qx;  Qz=randi([-10,10],z\_dim,z\_dim);  Qz=Qz'\*Qz;  A=randi([-10,10],5,x\_dim);  B=randi([-10,10],5,z\_dim);  c=A\*x\_real+B\*z\_real;  rho=0.001;  y\_dim=length(c); | %% the LOOP  x\_val=ones(x\_dim,1);z\_val=ones(z\_dim,1);y\_val=ones(y\_dim,1);  z\_k=z\_val; y\_k=y\_val;  N=1e2; % # iterations  x\_cost\_history=zeros(1,N);  z\_cost\_history=zeros(1,N);  constraint\_cost\_history=zeros(1,N);  for ii=1:1:N  rho=1/ii;  [x\_kp1] = minimize\_x(z\_k,y\_k,Qx,Qz,A,B,c,rho);  [z\_kp1] = minimize\_z(x\_kp1,y\_k,Qx,Qz,A,B,c,rho);  y\_kp1=y\_k + rho\*(A\*x\_kp1+B\*z\_kp1-c);  z\_k=z\_kp1; y\_k=y\_kp1;    x\_cost\_history(ii)=x\_kp1'\*Qx\*x\_kp1;  z\_cost\_history(ii)=z\_kp1'\*Qz\*z\_kp1;  constraint\_cost\_history(ii)=norm(A\*x\_kp1+B\*z\_kp1-c,2);  end |
| function [x\_kp1] = minimize\_x(z\_k,y\_k,Qx,Qz,A,B,c,rho)  % INPUTS z\_k,y\_k,Qx,Qz,A,B,c,rho  % OUTPUTS x\_kp1  x=sdpvar(size(Qx,1),1);  z=z\_k;  y=y\_k;  Lp\_xz\_y=(x'\*Qx\*x)+(z'\*Qz\*z)+y'\*(A\*x+B\*z-c)+(rho/2)\*norm((A\*x+B\*z-c),2)^2;  % DIAGNOSTIC = OPTIMIZE(Constraint,Objective,options)  options = sdpsettings('solver','mosek');  DIAGNOSTIC = optimize([],Lp\_xz\_y,options);  x\_kp1=value(x);  end | function [z\_kp1] = minimize\_x(x\_kp1,y\_k,Qx,Qz,A,B,c,rho)  % INPUTS xkp1,y\_k,Qx,Qz,A,B,c,rho  % OUTPUTS z\_kp1  z=sdpvar(size(Qz,1),1);  x=x\_kp1;  y=y\_k;  Lp\_xz\_y=(x'\*Qx\*x)+(z'\*Qz\*z)+y'\*(A\*x+B\*z-c)+(rho/2)\*norm((A\*x+B\*z-c),2)^2;  % DIAGNOSTIC = OPTIMIZE(Constraint,Objective,options)  options = sdpsettings('solver','mosek');  DIAGNOSTIC = optimize([],Lp\_xz\_y,options);  z\_kp1=value(z);  end |

a

|  |  |
| --- | --- |
| %% PRINTING THE RESULTS  disp('==============================');  disp(['norm(x\_real-x\_kp1,2) : ',num2str(norm(x\_real-x\_kp1,2))]);  disp(['norm(z\_real-z\_kp1,2) : ',num2str(norm(z\_real-z\_kp1,2))]);  disp(['cost x\_real:',num2str(x\_real'\*Qx\*x\_real)]);  disp(['cost z\_real:',num2str(z\_real'\*Qz\*z\_real)]);  disp(['cost x:',num2str(x\_kp1'\*Qx\*x\_kp1)]);  disp(['cost z:',num2str(z\_kp1'\*Qz\*z\_kp1)]);  disp(['ineq cost:',num2str(norm(A\*x\_kp1+B\*z\_kp1-c,2))]);  disp('==============================');  fig\_1=figure(1); fig\_1.Color=[1,1,1];  plot(1:1:N,x\_cost\_history,'r.'); hold on;  plot(1:1:N,z\_cost\_history,'b.'); hold on;  plot(1:1:N,constraint\_cost\_history,'k.'); hold on;  xlabel('iter');  ylabel('cost values');  legend('cost(x)','cost(z)','cost(CONSTRAINT)');  fig\_1.CurrentAxes.FontSize=15; | Chart  Description automatically generated |