```
condition = true;
while( condition == true)

B = A(1:end, J);
K = setdiff(indices,J);
ak = A(1:end,K);

y = (inv(B.')) * transpose(c(J));
u = zeros(dimension,1);
u(K) = transpose(c(K)) - ak.'*y;
condition = min(u) < 0;

%initialization
a = 0;
r = 0;
if(condition)
    a = u<0; %logical vector
    r = find(a,1);%index of first non-zero value aka first index where u<0
    d = zeros(dimension,1);
    d(J) = (inv(B.'))* A(1:end,r);

    test = d(d ~= Inf);
    condition = (max(test)>0);
    if(condition)
        xOverd = ones(dimension,1);
        xOverd = xOverd ./ 0; %set all to infty
        d(d<0) = Inf; % if negative set to infty
        xOverd(J) = x(J) ./ d(J);
        xOverd(xOverd == 0) = Inf; %if inf/inf gets a not desired 0 in there
        minRatio  = min(xOverd(J));
        s = find(xOverd == minRatio);
        s = min(s);
        %blands rule;
        %all the Inf were there so there not considered in when taking the
        %minRatio, a NaN would probably have been smarter in retrospect

        %construct new vector
        d(d == Inf) = 0;
        x(J) = x(J) - minRatio * d(J);
        x(r)  = minRatio;
        J = union(J,r);
        J = setdiff(J,s);
        x(K) = 0;
    else
        Disp('The problem is unbounded')
    end

else % i.e. u greater than zero
    disp('The minimizer is: \n')
end
end
```

```
The minimizer is: \n
```

Published with MATLAB® R2014a