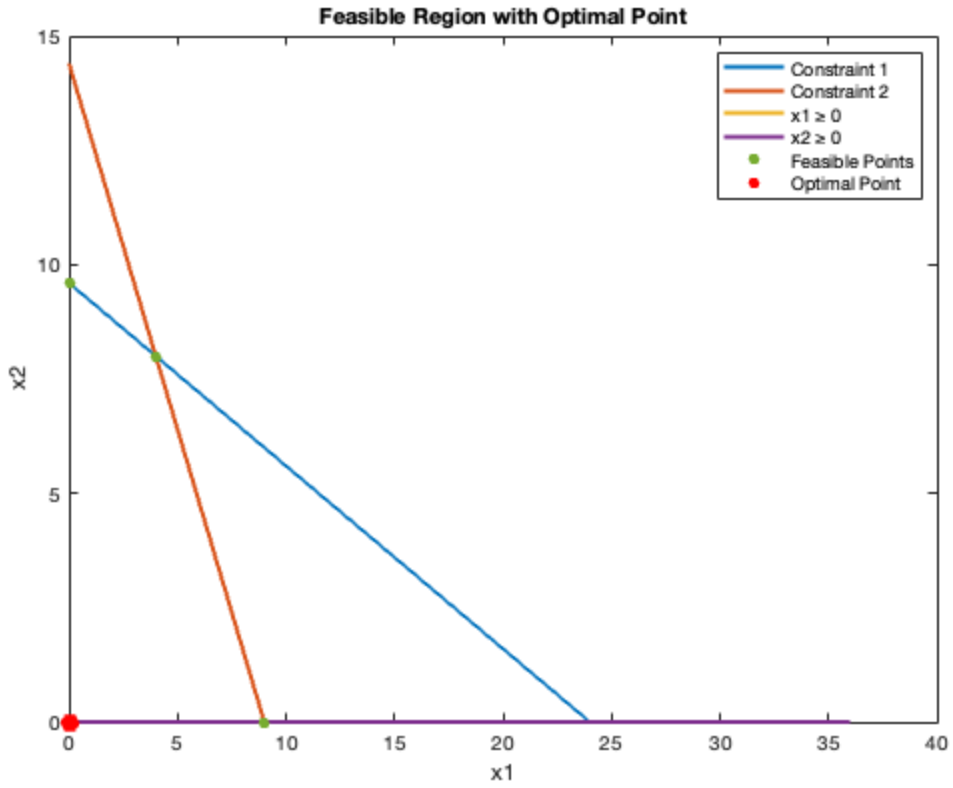
```
clc;
clear all;
format short;

% Input parameters
c = [40, 24]; % Cost in objective function
A = [20, 50; 80, 50; -1, 0; 0, -1]; % Adding negative constraints for non-
negativity
B = [480; 720; 0; 0];
n = size(A,1);
x1 = 0:0.01:max(B)/20; % Adjusted for better plotting scale

% Drawing the lines and finding feasible region
for i = 1:n
    if all(A(i,:) == 0)
        continue; % Skip trivial constraints
    end
    y(i,:) = (B(i) - A(i,1) * x1) / A(i,2);
    y(i,:) = max(0, y(i,:)); % Non-negativity
    plot(x1, y(i,:), 'linewidth', 2);
    hold on;
end

% Finding intersection points
pt = [];
for i = 1:size(A,1)
    for j = i+1:size(A,1)
        A3 = [A(i,:); A(j,:)];
        B3 = [B(i); B(j)];
        if rank(A3) == 2 % Ensure the system of equations is solvable
            X3 = A3 \ B3;
            if all(X3 >= 0) % Non-negativity check
                pt = [pt; X3'];
            end
        end
    end
end

% Unique and feasible points
X = unique(pt, 'rows');

% Check each point for all constraints
feasible = all(A * X' - B <= 0);
X = X(feasible,:);

% Evaluate Objective Function
obj_val = c * X';
[value, ind] = min(obj_val);
Optimal = [X(ind,:), value];

% Scatter plot of feasible points and optimal point
scatter(X(:,1), X(:,2), 'filled');
```

```
hold on;
scatter(Optimal(1), Optimal(2), 100, 'red', 'filled');
hold off;

% Setting the axes
xlabel('x1');
ylabel('x2');
title('Feasible Region with Optimal Point');
legend('Constraint 1', 'Constraint 2', 'x1 ≥ 0', 'x2 ≥ 0', 'Feasible Points',
'Optimal Point');

% Displaying the optimal solution
disp('Optimal Solution:');
disp(['x1 = ', num2str(Optimal(1))]);
disp(['x2 = ', num2str(Optimal(2))]);
disp(['Minimized Value = ', num2str(Optimal(3))]);
```

```
Optimal Solution:
x1 = 0
x2 = 0
Minimized Value = 0
```



Published with MATLAB® R2023b