Kabahemba Joanitah

2023/U/MMU/BCS/01667

1

import libraries from pulp import * import numpy as np import matplotlib.pyplot as plt define linear problem problem= LpProblem(name="resource$_M inimization", sense = LpMinimize$)

define decision variables x=LpVariable("X",0) y=LpVariable("Y",0)

define the objective problem+= 4*x + 5*y ,"objective"

define constraints problem+= 2*x + 3*y ¿= 10,"CPU" problem+= x + 2*y ¿= 5,"Memory" problem+= 3*x + y ¿= 8,"Storage"

solve and show problem.solve()

display results print("OPTIMUM SOLUTION")

print(f"X:x.varValue") print(f"Y:y.varValue")

print(f"Minimum$_c ost : problem.objective.value()$")

the graph x array

x=np.linspace(0,16,20)

convert constraints to inequalities y1=(10 -2*x)/3 y2=(5 - x)/2 y3=(8-3*x)

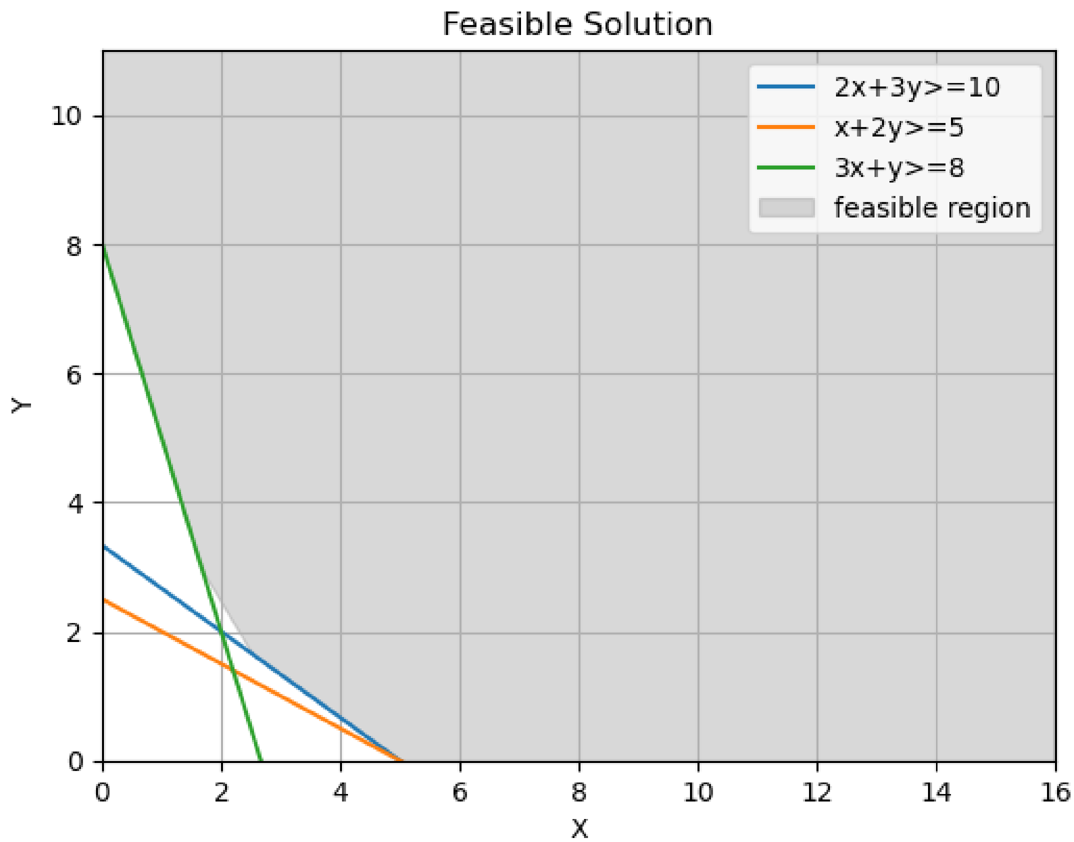
plot constraints plt.plot(x,y1 ,label="2x+3y¿=10") plt.plot(x,y2 ,label="x+2y¿=5") plt.plot(x,y3, label="3x+y¿=8")

plotting the feasible region y4=np.maximum.reduce([y1,y2,y3]) upper boundary of the feasible region

plt.fill$_b etween(x, y4, 11, color =' grey', alpha = 0.3, label = "feasibleregion"$)

axis limits and labels plt.xlim(0,16) plt.ylim(0,11) plt.xlabel("X") plt.ylabel("Y") plt.legend() plt.title("Feasible Solution") plt.grid()

plt.show()



import libraries from pulp import *

define linear problem problem= LpProblem(name="time$_M inimization", sense = LpMinimize$)

define decision variables x=LpVariable(name="X",lowBound=0) y=LpVariable(name="Y",lowBound=0)

define the objective problem+= 4*x + 5*y ,"objective"

define constraints problem+= 2*x + 3*y ¡= 20,"Server 1" problem+= 4*x + 2*y ¡= 15,"Server 2"

solve and show problem.solve()

display results print("OPTIMUM SOLUTION")

print(f"X:x.varValue") print(f"Y:y.varValue")

print(f"Minimum$_t ime : problem.objective.value()$")

the graph x array

x=np.linspace(0,16,20)

convert constraints to inequalities y1=(20 -2*x)/3 y2=(15 - 4*x)/2

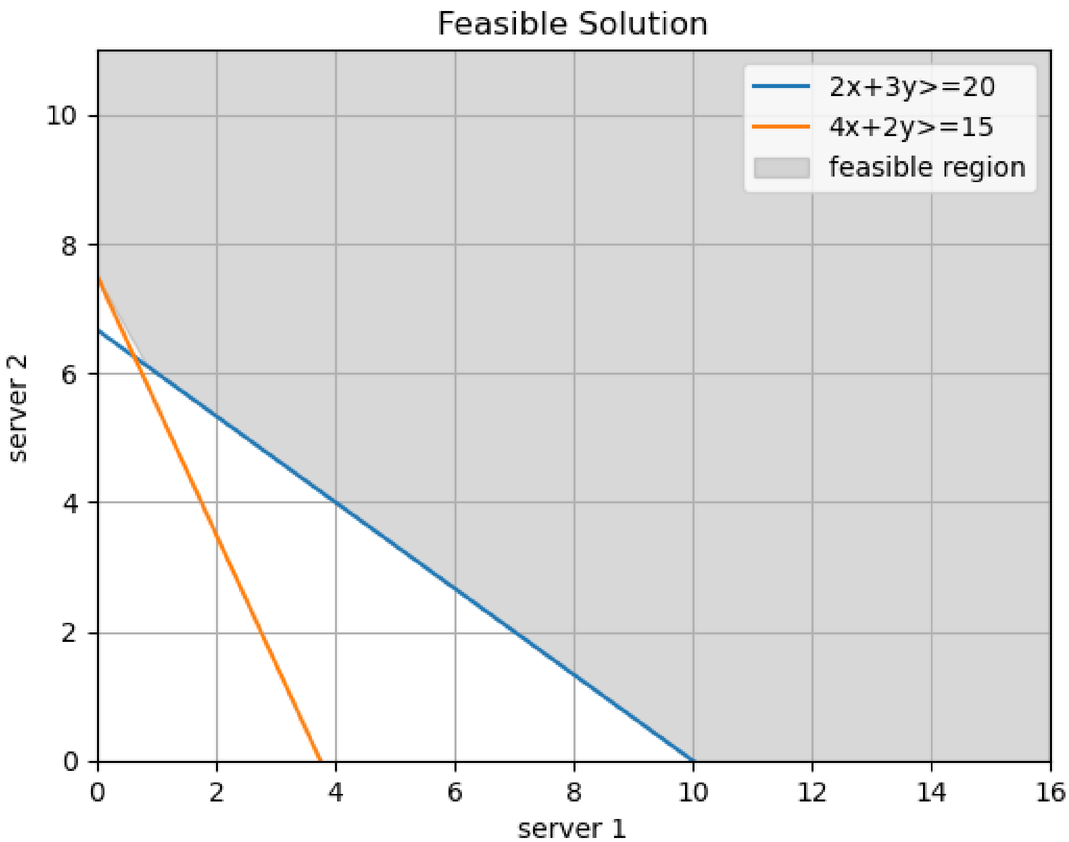
plot constraints plt.plot(x,y1 ,label="2x+3y¿=20") plt.plot(x,y2 ,label="4x+2y¿=15")

plotting the feasible region y3=np.maximum.reduce([y1,y2,]) upper boundary of the feasible region

plt.fill$_b etween(x, y3, 11, color =' grey', alpha = 0.3, label = "feasibleregion")$

axis limits and labels plt.xlim(0,16) plt.ylim(0,11) plt.xlabel("server 1") plt.ylabel("server 2") plt.legend() plt.title("Feasible Solution") plt.grid()

plt.show



import libraries from pulp import *

define linear problem problem= LpProblem(name="energy$_M inimization", sense = LpMinimize$)

define decision variables x=LpVariable(name="X",lowBound=0) y=LpVariable(name="Y",lowBound=0)

define the objective problem+= 3*x + 2*y ,"objective"

define constraints problem+= 2*x + 3*y ¿= 15,"CPU Aloocation" problem+= 4*x + 2*y ¿= 10,"Memory Allocation"

solve and show problem.solve()

display results print("OPTIMUM SOLUTION")

print(f"X:x.varValue") print(f"Y:y.varValue")

print(f"Minimum$_e nergy : problem.objective.value()$")

the graph x array

x=np.linspace(0,16,20)

convert constraints to inequalities y1=(15 -2*x)/3 y2=(10 - 4*x)/2

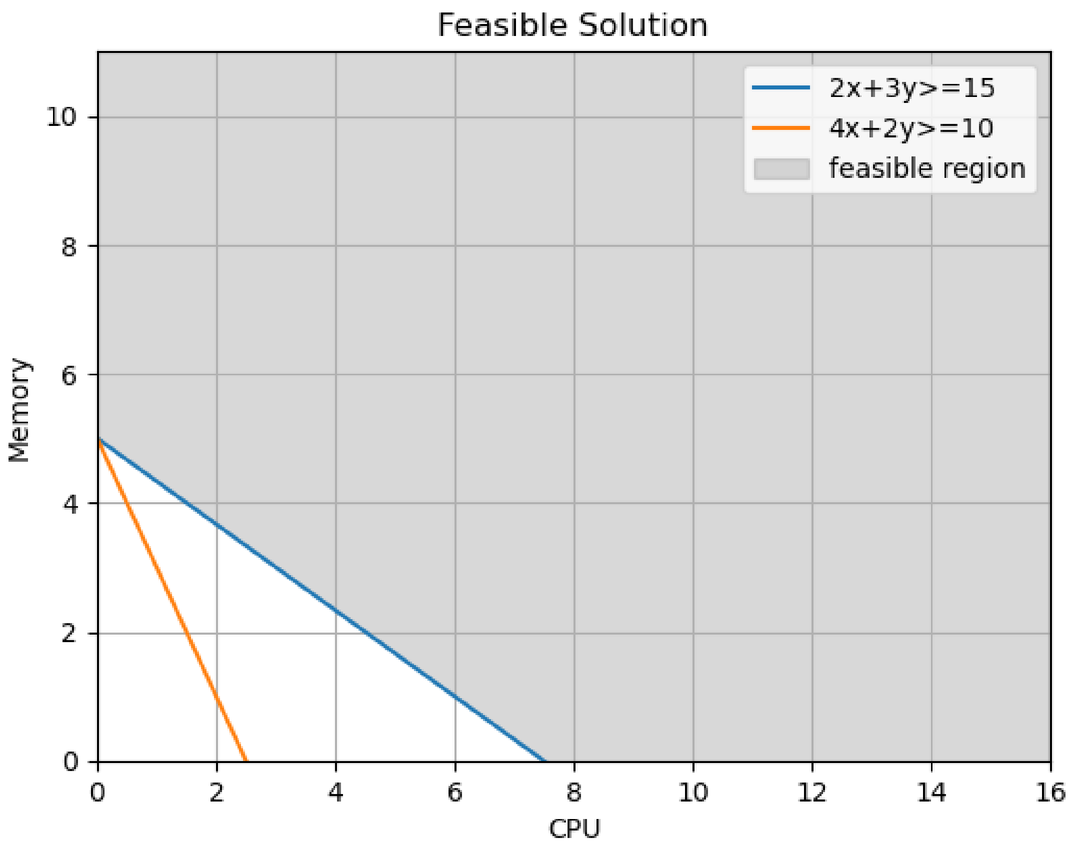
plot constraints plt.plot(x,y1 ,label="2x+3y¿=15") plt.plot(x,y2 ,label="4x+2y¿=10")

plotting the feasible region y3=np.maximum.reduce([y1,y2,]) upper boundary of the feasible region

plt.fill$_{b}etween(x, y3, 11, color =' grey', alpha = 0.3, label = "feasibleregion")$

axis limits and labels plt.xlim(0,16) plt.ylim(0,11) plt.xlabel("CPU") plt.ylabel("Memory") plt.legend() plt.title("Feasible Solution") plt.grid()

plt.show



import libraries from pulp import *

define linear problem problem= LpProblem(name="resource$_{M}inimization", sense = LpMinimize)$

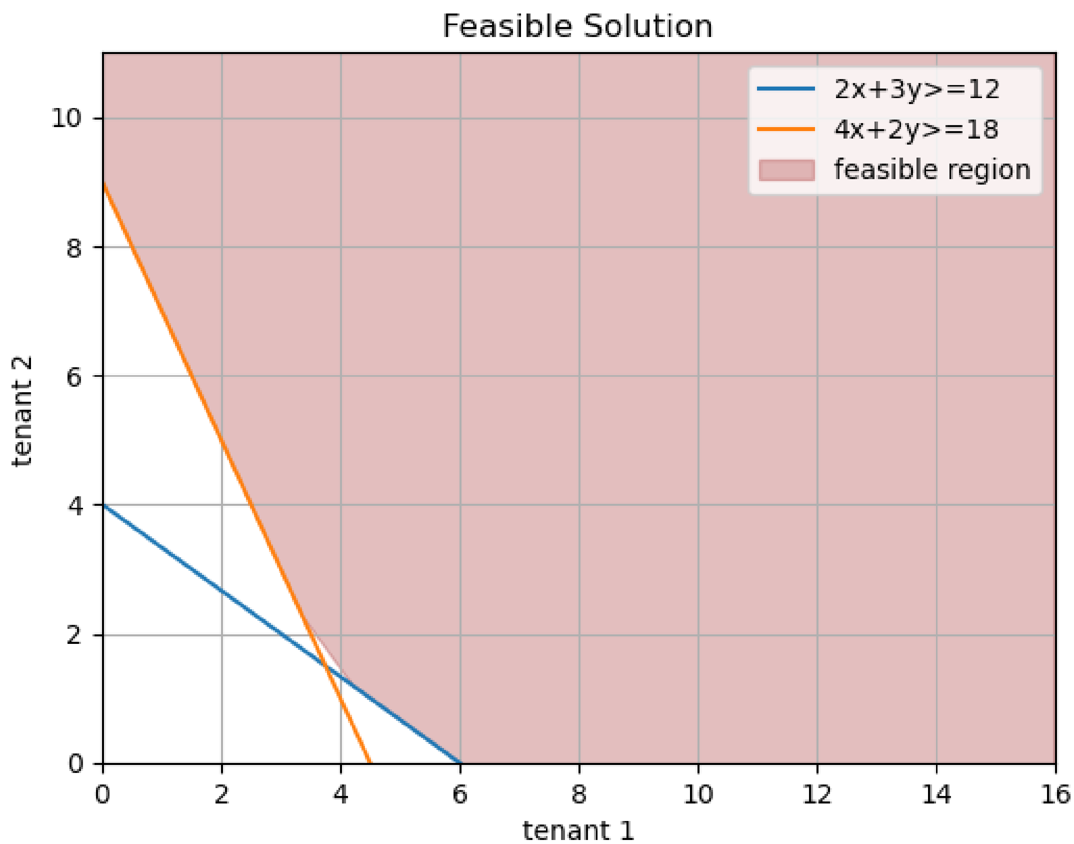
define decision variables x=LpVariable(name="X",lowBound=0) y=LpVariable(name="Y",lowBound=0)

define the objective problem+= 5*x + 4*y ,"objective"

define constraints problem+= 2*x + 3*y ¿= 12,"tenant 1" problem+= 4*x + 2*y ¿= 18,"tenant 2"

solve and show problem.solve()

display results print(”OPTIMUM SOLUTION”)
print(f”X:x.varValue”) print(f”Y:y.varValue”)
print(f”Minimum$_r esource : problem.objective.value()$”)
the graph x array
x=np.linspace(0,16,20)
convert constraints to inequalities y1=(12 -2*x)/3 y2=(18 - 4*x)/2
plot constraints plt.plot(x,y1 ,label=”2x+3y¿=12”) plt.plot(x,y2 ,label=”4x+2y¿=18”)
plotting the feasible region y3=np.maximum.reduce([y1,y2,]) upper boundary of the feasible region
plt.fill$_b etween(x, y3, 11, color =' brown', alpha = 0.3, label = ” feasibleregion$”)
axis limits and labels plt.xlim(0,16) plt.ylim(0,11) plt.xlabel(”tenant 1”) plt.ylabel(”tenant 2”) plt.legend() plt.title(”Feasible Solution”) plt.grid()
plt.show



import libraries from pulp import * import numpy as np import matplotlib.pyplot as plt define linear problem problem= LpProblem(name=”cost$_M inimization”, sense = LpMinimize$)

define decision variables x1=LpVariable(name="quantity of product1",lowBound=0) x2=LpVariable(name="quantity of product2",lowBound=0) x3=LpVariable(name="quantity of product3",lowBound=0)

define the objective problem+= 5*x1 + 3*x2 + 4*x3 ,"objective"

define constraints problem+= 2*x1 + 3*x2 +x3 ¡= 1000,"raw material" problem+= 4*x1 + 2*x2 + 5*x3 ¡= 120,"labor hours"

problem+= x1¿=200 problem+= x2¿=300 problem+= x3¿=150

solve and show problem.solve()

display results print("OPTIMUM SOLUTION")

print(f"quantity of product1:x1.varValue") print(f"quantity of product2:x2.varValue") print(f"quantity of product3:x3.varValue")

print(f"Maximise$_{c}ost : problem.objective.value()$")

plotting the graph

create a meshgrid for x1,x2,and x3 x1$_{v}als = np.linspace(0, 400, 50)x2_{v}als = np.linspace(0, 400, 50)x1_{g}rid, x2_{g}rid = np.meshgrid(x1_{v}als, x2_{v}als)$

calculate the corresponding z-values(objective function) z$_{v}als = 5 * x1_{g}rid + 3 * x2_{g}rid + 4 * (1950 - x1_{g}rid - x2_{g}rid)$

create the 3d plot fig=plt.figure(figsize=(10,6)) ax=fig.add$_{s}ubplot(111, projection =' 3d')$

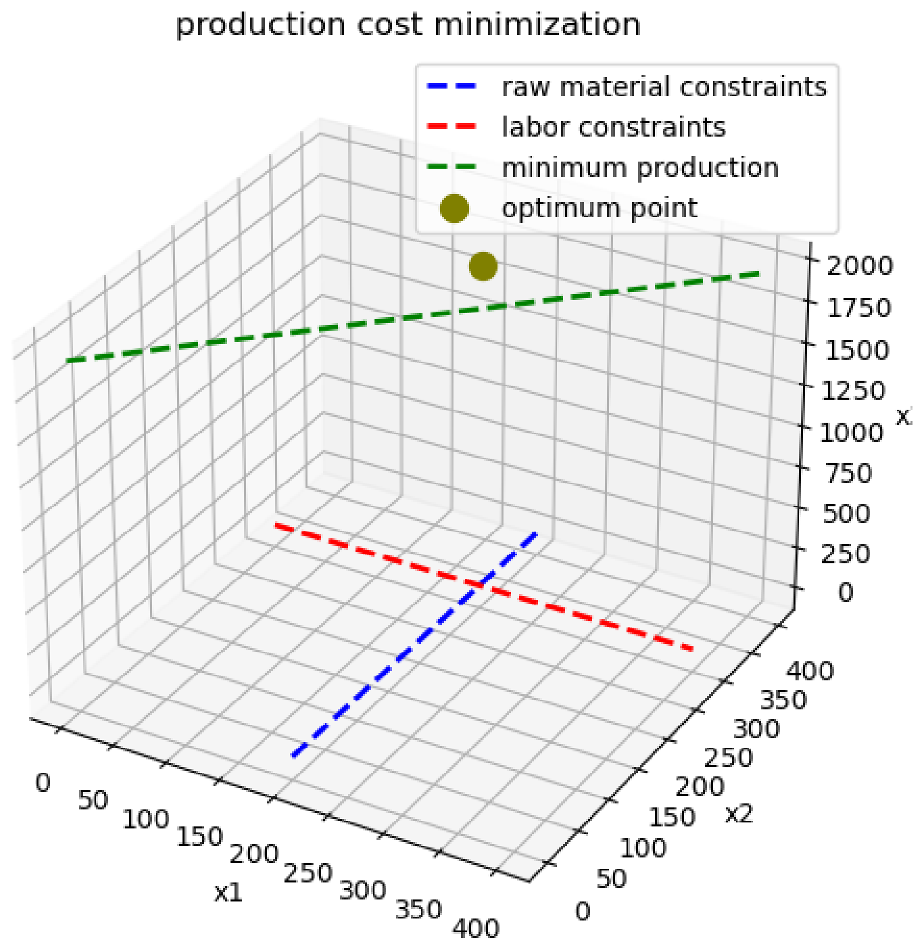
plot the feasible region(constraints) ax.plot([200,200],[0,400],[0,0],color='blue',linestyle='–',linewidth=2,label='raw material constraints') ax.plot([0,400],[300,300],[0,0],color='red',linestyle='–',linewidth=2,label='labor constraints') ax.plot([0,400],[0,400],[1950,1950],color='green',linestyle='–',linewidth=2,label='minimum production')

highlight the minimum point optimum$_{x}1 = 200 optimum_{x}2 = 300 optimum_{z} = 1950 ax.scatter(optimum_{x}1, optimum_{x}2, optimum_{z}, color =' olive', s = 100, label =' optimumpoint')$

set labels and title ax.set$_{x}label('x1')ax.set_{y}label('x2')ax.set_{z}label('x3')ax.set_{t}itle('productioncostminimiz$

add a legend ax.legend()

show the plot plt.show()



import libraries from pulp import *

define linear problem problem= LpProblem(name="investment$_{M}aximization", sense = LpMaximize$)

define decision variables x1=LpVariable(name="X1",lowBound=0) x2=LpVariable(name="X2",lowBound= x3=LpVariable(name="X3",lowBound=0)

define the objective problem+= 0.08*x1 + 0.1*x2 + 0.12*x3 ,"objective"

define constraints problem+= 2*x1 + 3*x2 +x3 ¡= 10000,"budget constrait" problem+= x1¿=2000," stock A" problem+= x2¿=1500 ," stock B" problem+= x3¿=1000," stock C"

solve and show problem.solve()

display results print("OPTIMUM SOLUTION")

print(f"X1:x1.varValue") print(f"X2:x2.varValue") print(f"X3:x3.varValue")

print(f"Maximise$_{i}nvestment : problem.objective.value()$")

plotting the graph

create a meshgrid for A,B,and C A$_{v}als = np.linspace(0, 2500, 50)B_{v}als =$

$np.linspace(0, 2000, 50) A_grid, B_grid = np.meshgrid(A_vals, B_vals)$

calculate the corresponding z-values(ROI function) $C_vals = (10000 - 2 * A_grid - 3 * B_grid) budget constraint ROI_vals = 0.08 * A_grid + 0.1 * B_grid + 0.12 * C_vals$

create the 3d plot fig=plt.figure(figsize=(10,6)) ax=fig.add$_subplot(111, projection =' 3d')$

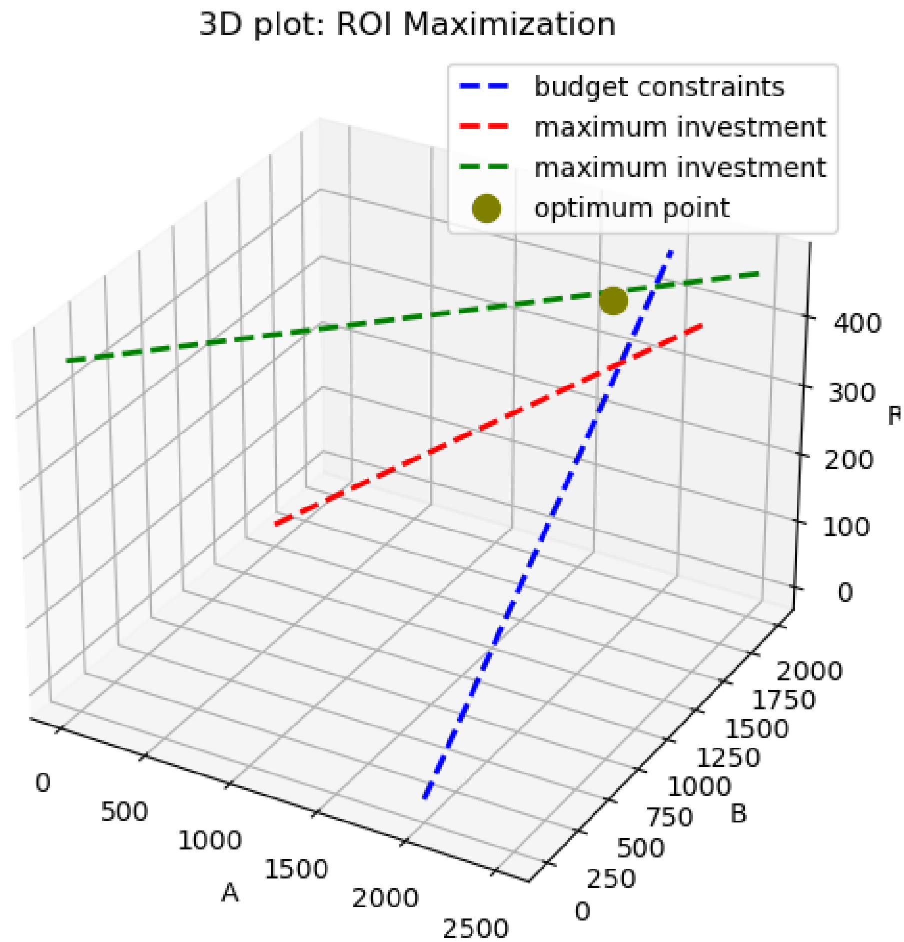
plot the feasible region(constraints) ax.plot([2000,2000],[0,2000],[0,470],color='blue',linestyle='–',linewidth=2,label='budget constraints') ax.plot([0,2500],[1500,1500],[0,470],color='red',linestyle='–',linewidth=2,label='maximum investment') ax.plot([0,2500],[0,2000],[470,470],color='green',linestyle='–',linewidth=2,label='maximum investment')

highlight the minimum point optimum$_A = 2000 optimum_B = 1500 optimum_ROI = 470 ax.scatter(optimum_A, optimum_B, optimum_ROI, color =' olive', s = 100, label =' optimumpoint')$

set labels and title ax.set$_xlabel('A') ax.set_ylabel('B') ax.set_zlabel('ROI') ax.set_title('3Dplot : ROIMaximization')$

add a legend ax.legend()

show the plot plt.show()



import libraries from pulp import *

define linear problem problem= LpProblem(name=”cost$_{M}inimization”, sense = LpMinimize$)

define decision variables x1=LpVariable(name=”X1”,lowBound=0) x2=LpVariable(name=”X2”,lowBound=

define the objective problem+= 3*x1 + 2*x2 ,”objective”

define constraints problem+= 2*x1 + x2 ¿= 20,”item 1” problem+= 3*x1 + 2*x2 ¿= 25,”item 2”

solve and show problem.solve()

display results print(”OPTIMUM SOLUTION”)

print(f”X1:x1.varValue”) print(f”X2:x2.varValue”)

print(f”Minimum$_{c}ost : problem.objective.value()$”)

the graph x array

x1=np.linspace(0,16,20)

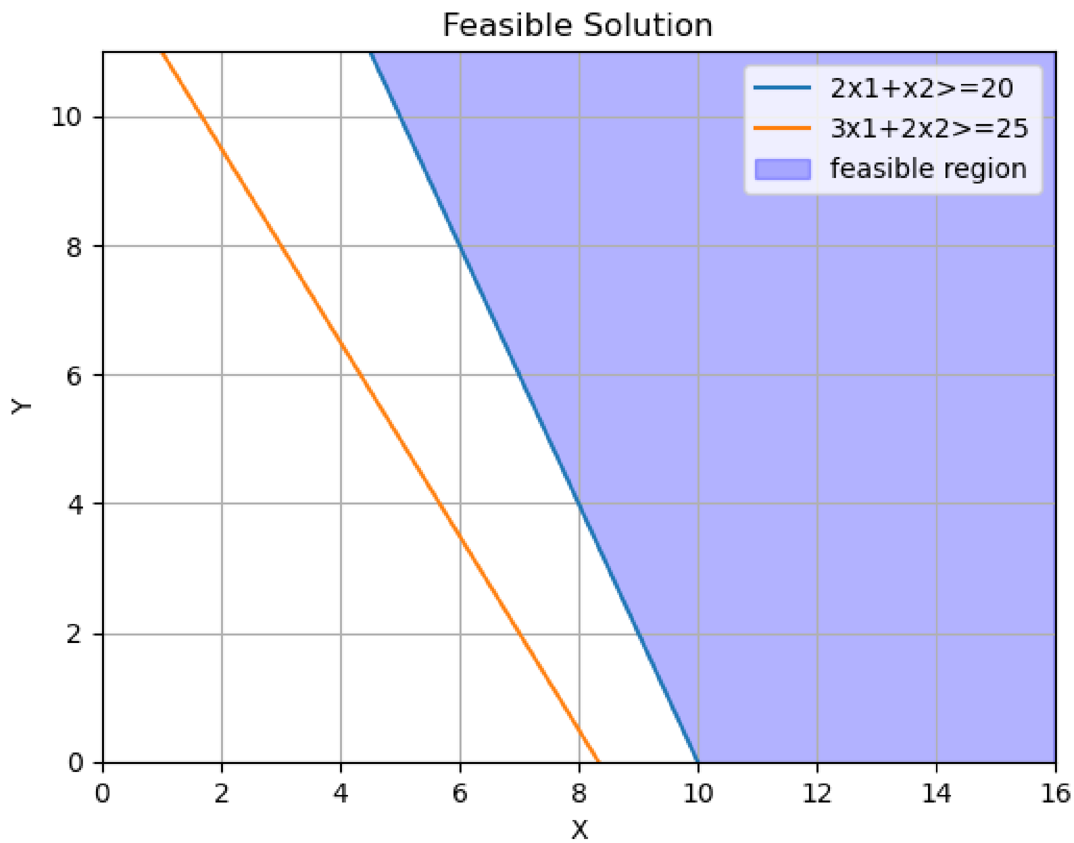
convert constraints to inequalities y1=(20 -2*x1) y2=(25 - 3*x1)/2

plot constraints plt.plot(x1,y1 ,label=”2x1+x2¿=20”) plt.plot(x1,y2 ,label=”3x1+2x2¿=25”) plotting the feasible region y3=np.maximum.reduce([y1,y2,]) upper boundary of the feasible region

plt.fill$_{b}etween(x1, y3, 11, color =' blue', alpha = 0.3, label = "feasibleregion")$

axis limits and labels plt.xlim(0,16) plt.ylim(0,11) plt.xlabel("X") plt.ylabel("Y") plt.legend() plt.title("Feasible Solution") plt.grid()

plt.show()



import libraries from pulp import *

define linear problem problem= LpProblem(name="profit$_{M}aximization", sense = LpMaximize)$

define decision variables x1=LpVariable(name="X1",lowBound=0) x2=LpVariable(name="X2",lowBound=

define the objective problem+= 5*x1 + 3*x2 ,"objective"

define constraints problem+= 2*x1 + 3*x2 ¡= 60,"product 1" problem+= 4*x1 + 2*x2 ¡= 80,"product 2"

solve and show problem.solve()

display results print("OPTIMUM SOLUTION")

print(f"X1:x1.varValue") print(f"X2:x2.varValue") print(f"Maximum$_profit : problem.objective.value()$")

the graph x array

x1=np.linspace(0,25,20)

convert constraints to inequalities y1=(60 -2*x1)/3 y2=(80 - 4*x1)/2

plot constraints plt.plot(x1,y1 ,label="2x1+3x2¿=60") plt.plot(x1,y2 ,label="4x1+2x2¿=80")

plotting the feasible region y3=np.maximum.reduce([y1,y2,]) upper boundary of the feasible region

plt.fill$_between(x1, y3, 20, color =' olive', alpha = 0.3, label = "feasibleregion")$

axis limits and labels plt.xlim(0,25) plt.ylim(0,11) plt.xlabel("Labor") plt.ylabel("Raw material") plt.legend() plt.title("Feasible Solution") plt.grid()

plt.show()