Practical-1

!pip install pulp

from pulp import \*

import matplotlib.pyplot as plt

import numpy as np

# Optimal Solution

model1 = pulp.LpProblem('linear\_programming', LpMaximize)

solver1 = getSolver('PULP\_CBC\_CMD')

x1 = LpVariable('x1', lowBound = 0, cat = 'çontinuous')

x2 = LpVariable('x2', lowBound = 0, cat = 'continuous')

# Objective Function

model1 += 25\*x1 + 20\*x2

# Constraints

model1 += 5\*x1 + 3\*x2 <= 500

model1 += 5\*x1 + 5\*x2 <= 540

model1 += x1 >= 0

model1 += x2 >= 0

results = model1.solve(solver = solver1)

if LpStatus[results] == 'Optimal':

 if value(x1) == 0 or value(x2) == 0:

  print('The solution is degenerate')

 else:

  print('The solution is optimal')

  print(f'Objective value: z\* = {value(model1.objective)}')

  print(f'Solution: x1\* = {value(x1)}, x2\* = {value(x2)}')

elif LpStatus[results] == 'Unbounded':

  print('The solution is unbounded')

  print(f'Objective value: z\* = {value(model1.objective)}')

  print(f'Solution: x1\* = {value(x1)}, x2\* = {value(x2)}')

elif LpStatus[results] == 'Infeasible':

  print('The solution is infeasible')

  print(f'Objective value: z\* = {value(model1.objective)}')

  print(f'Solution: x1\* = {value(x1)}, x2\* = {value(x2)}')

# Graphical Representation

x = np.arange(0, 500)

plt.plot(x, (500 - 5 \* x)/3, label = '5\*x1 + 3\*x2 <= 500')

plt.plot(x, (540 - 5 \* x)/5, label= '5\*x1 + 5\*x2 <= 540')

x = [0, 100, 88, 0]

y = [0, 0, 20, 108]

plt.fill(x, y, 'grey')

plt.text(30, 30, 'Feasible \n Region', size = '11')

plt.annotate('Optimal \n solution\n(88, 20)', xy = (88, 20))

plt.xlabel("x\_1")

plt.ylabel("x\_2")

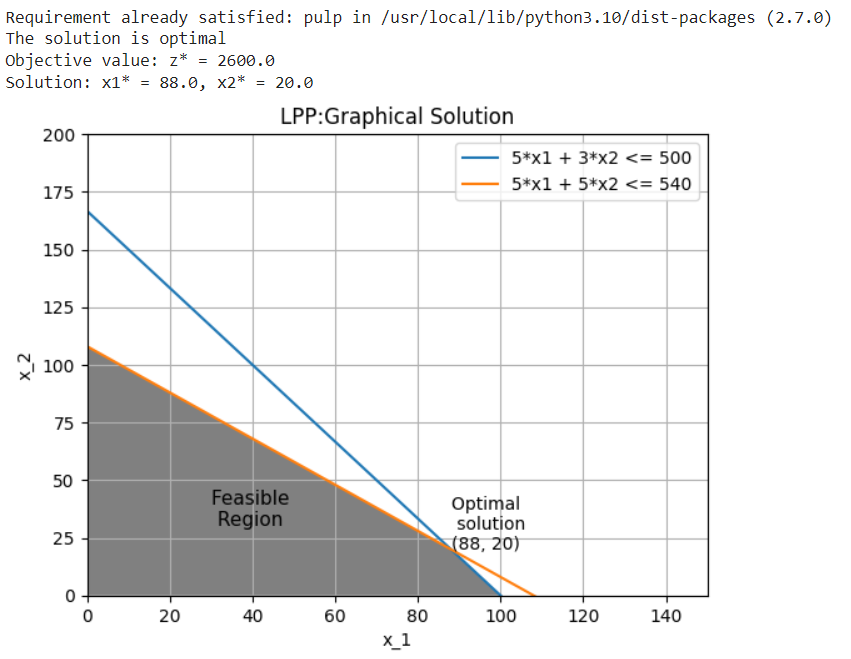
plt.title('LPP:Graphical Solution')

plt.axis([0, 150, 0, 200])

plt.grid(True)

plt.legend()

plt.show()



# Unbounded Solution

model2 = pulp.LpProblem('linear\_programming', LpMaximize)

solver2 = getSolver('PULP\_CBC\_CMD')

x1 = LpVariable('x1', lowBound = 0, cat = 'çontinuous')

x2 = LpVariable('x2', lowBound = 0, cat = 'continuous')

# Objective Function

model2 += 5\*x1 + 4\*x2

# Constraints

model2 += x1 - 2\*x2 <= 1

model2 += x1 + 2\*x2 >= 100

model2 += x1 >= 0

model2 += x2 >= 0

results = model2.solve(solver = solver2)

if LpStatus[results] == 'Optimal':

 if value(x1) == 0 or value(x2) == 0:

  print('The solution is degenerate')

 else:

  print('The solution is optimal')

  print(f'Objective value: z\* = {value(model2.objective)}')

  print(f'Solution: x1\* = {value(x1)}, x2\* = {value(x2)}')

elif LpStatus[results] == 'Unbounded':

  print('The solution is unbounded')

  print(f'Objective value: z\* = {value(model2.objective)}')

  print(f'Solution: x1\* = {value(x1)}, x2\* = {value(x2)}')

elif LpStatus[results] == 'Infeasible':

  print('The solution is infeasible')

  print(f'Objective value: z\* = {value(model2.objective)}')

  print(f'Solution: x1\* = {value(x1)}, x2\* = {value(x2)}')

# Graphical Representation

x = np.arange(0, 101)

plt.plot(x, (-1 + x)/2, label = 'x1 - 2\*x2 <= 1')

plt.plot(x, (100 - x)/2, label= 'x1 + 2\*x2 >= 100')

x = [0, 50.5, 100, 100, 0]

y = [50, 24.75, 49.5, 100, 100]

plt.fill(x, y, 'grey')

plt.text(45, 80, 'Feasible \n Region', size = '11')

plt.annotate('Unbounded \n solution', xy = (50.5, 24.75))

plt.xlabel("x\_1")

plt.ylabel("x\_2")

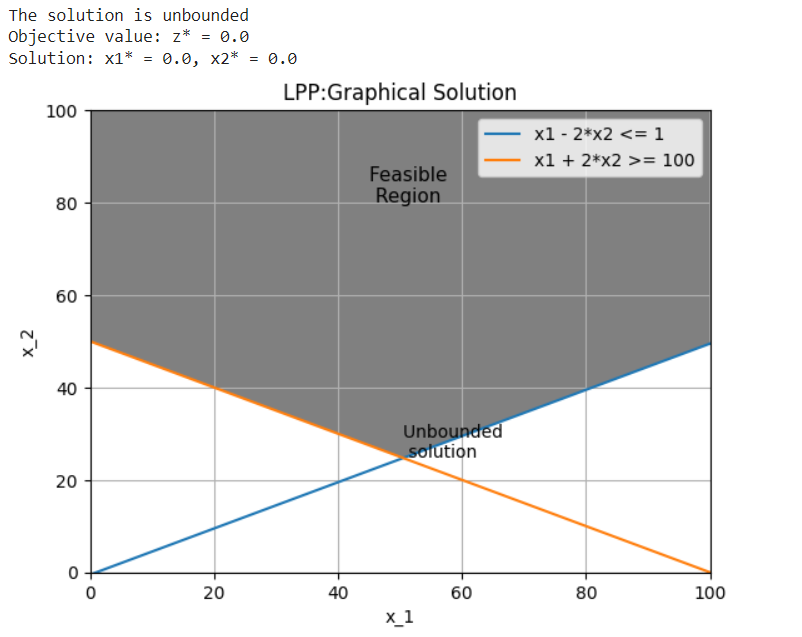
plt.title('LPP:Graphical Solution')

plt.axis([0, 100, 0, 100])

plt.grid(True)

plt.legend()

plt.show()



# Infeasible Solution

model3 = pulp.LpProblem('linear\_programming', LpMinimize)

solver3 = getSolver('PULP\_CBC\_CMD')

x1 = LpVariable('x1', lowBound = 0, cat = 'çontinuous')

x2 = LpVariable('x2', lowBound = 0, cat = 'continuous')

# Objective Function

model3 += 200\*x1 + 300\*x2

# Constraints

model3 += 2\*x1 + 3\*x2 >= 1200

model3 += x1 + x2 <= 400

model3 += 2\*x1 + 3/2\*x2 >= 900

model3 += x1 >= 0

model3 += x2 >= 0

results = model3.solve(solver = solver3)

if LpStatus[results] == 'Optimal':

 if value(x1) == 0 or value(x2) == 0:

  print('The solution is degenerate')

 else:

  print('The solution is optimal')

  print(f'Objective value: z\* = {value(model3.objective)}')

  print(f'Solution: x1\* = {value(x1)}, x2\* = {value(x2)}')

elif LpStatus[results] == 'Unbounded':

  print('The solution is unbounded')

  print(f'Objective value: z\* = {value(model3.objective)}')

  print(f'Solution: x1\* = {value(x1)}, x2\* = {value(x2)}')

elif LpStatus[results] == 'Infeasible':

  print('The solution is infeasible')

  print(f'Objective value: z\* = {value(model3.objective)}')

  print(f'Solution: x1\* = {value(x1)}, x2\* = {value(x2)}')

# Graphical Representation

x = np.arange(0, 1000)

plt.plot(x, (1200 - 2 \* x)/3, label = '2\*x1 + 3\*x2 >= 1200')

plt.plot(x, (400 - x), label= 'x1 + x2 <= 400')

plt.plot(x, (900 - 2 \* x)/1.5, label= '2\*x1 + 3/2\*x2 >= 900')

x = [0, 400, 0]

y = [0, 0, 400]

plt.fill(x, y, 'orange')

x = [0, 600, 600, 0]

y = [400, 0, 600, 600]

plt.fill(x, y, 'blue')

x = [0, 450, 600, 600, 0]

y = [600, 0, 0, 600, 600]

plt.fill(x, y, 'lightgreen')

plt.text(300, 400, 'No Feasible \n Region', size = '11')

plt.annotate('Infeasible \n solution', xy = (250, 150))

plt.xlabel("x\_1")

plt.ylabel("x\_2")

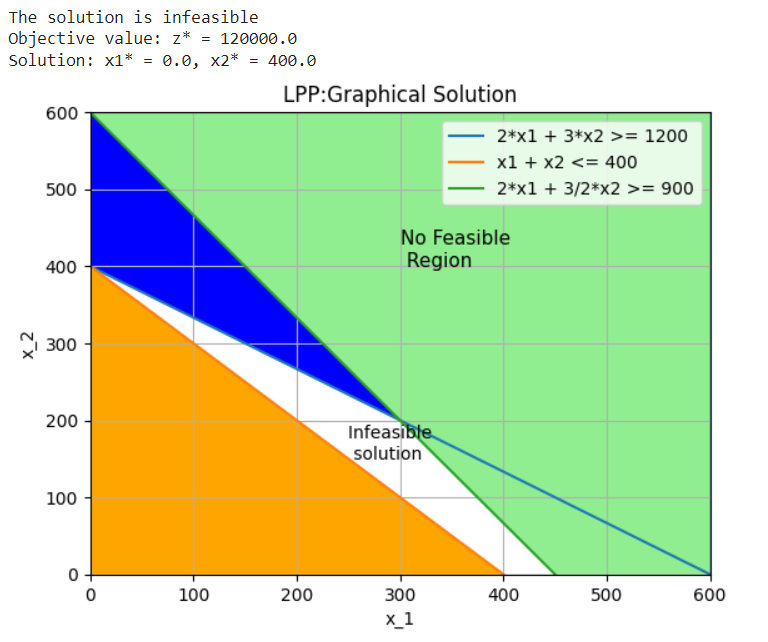
plt.title('LPP:Graphical Solution')

plt.axis([0, 600, 0, 600])

plt.grid(True)

plt.legend()

plt.show()



# Degenerate Solution

model5 = pulp.LpProblem('linear\_programming', LpMaximize)

solver5 = getSolver('PULP\_CBC\_CMD')

x1 = LpVariable('x1', lowBound = 0, cat = 'çontinuous')

x2 = LpVariable('x2', lowBound = 0, cat = 'continuous')

# Objective Function

model5 += 3\*x1 + 9\*x2

# Constraints

model5 += x1 + 4\*x2 <= 8

model5 += x1 + 2\*x2 <= 4

model5 += x1 >= 0

model5 += x2 >= 0

results = model5.solve(solver = solver5)

if LpStatus[results] == 'Optimal':

 if value(x1) == 0 or value(x2) == 0:

  print('The solution is degenerate')

 else:

  print('The solution is optimal')

  print(f'Objective value: z\* = {value(model5.objective)}')

  print(f'Solution: x1\* = {value(x1)}, x2\* = {value(x2)}')

elif LpStatus[results] == 'Unbounded':

  print('The solution is unbounded')

  print(f'Objective value: z\* = {value(model5.objective)}')

  print(f'Solution: x1\* = {value(x1)}, x2\* = {value(x2)}')

elif LpStatus[results] == 'Infeasible':

  print('The solution is infeasible')

  print(f'Objective value: z\* = {value(model5.objective)}')

  print(f'Solution: x1\* = {value(x1)}, x2\* = {value(x2)}')

# Graphical Representation

x = np.arange(0, 6)

plt.plot(x, (8 - x)/4, label = 'x1 + 4\*x2 <= 8')

plt.plot(x, (4 - x)/2, label= 'x1 + 2\*x2 <= 4')

x = [0, 4, 0]

y = [0, 0, 2]

plt.fill(x, y, 'grey')

plt.text(1, 0.5, 'Feasible \n Region', size = '11')

plt.text(2.5, 2, 'Degenerate \n Solution', size = '11')

plt.annotate('Optimal \n solution 1\n(0, 2)', xy = (0, 2))

plt.xlabel("x\_1")

plt.ylabel("x\_2")

plt.title('LPP:Graphical Solution')

plt.axis([0, 5, 0, 3])

plt.grid(True)

plt.legend()

plt.show()

