FINAL LAB TEST MDSC-103P

1. Wilson Problem: Wilson Manufacturing produces both baseballs and softballs, which it wholesales to vendors around the country. Its facilities permit the manufacture of a maximum of 500 dozen baseballs and a maximum of 500 dozen softballs each day. The cowhide covers for each ball are cut from the same processed cowhide sheets. Each dozen baseballs require five square feet of cowhide (including waste), whereas, one dozen softballs require six square feet of cowhide (including waste). Wilson has 3600 square feet of cowhide sheets available each day. Production of baseballs and softballs includes making the inside core, cutting and sewing the cover, and packaging. It takes about one minute to manufacture a dozen baseballs and two minutes to manufacture a dozen softballs. A total of 960 minutes is available for production daily. The prices for a dozen baseball and a dozen softball are 7 and 10 dollars respectively. Answer the following:

a) Formulate the problem in the Excel file and generate the sensitivity analysis.

Microsoft Excel 16.0 Sensitivity Report
Worksheet: [Book1]Sheet1
Report Created: 10-10-2023 13:35:50

Variable Cells

Cell	Name	Final Value	Reduced Cost	Objective Coefficient	Allowable Increase	Allowable Decrease
C4	x1	360	0	7	1.333333333	2
D4	x2	300	0	10	4	1.6

Constraints

Cell	Name	Final Value	Shadow Price	Constraint R.H. Side	Allowable Increase	Allowable Decrease
E5	c1	660	0	1000	1E+30	340
E6	c2	3600	1	3600	1200	720
E7	c3	960	2	960	240	240

b) Write on cost coefficient sensitivity analysis.

The cost sensitivity analysis tells us how the cost coefficient affect the optimality of the the given linear programming problem,the cost coefficient can be solved for an linear programming problem using an excel solver,from the cost coefficient analysis we can tell about the reduced cost and the shadow price , the shadow price tells us that at one unit of time how the materials used so that it makes an impact in the objective function
The shadow price is given by

Shadow Price
0
1
2

The reduced cost is

FINAL LAB TEST MDSC-103P

Reduced Cost
0
0

The allowable decrease for the x1 is 2 and the allowable increase for the x1 is 1.333
The allowable decrease for the x2 is 1.6 an allowable increase for the x2 is 4
X1 -baseballs
X2-softballs

The cost function tells us what impact will it make if there is like for example 10 per cent increase the material how it affect the optimality

If a new material is introduced how it effect the optimality etc

c) **Write on Right Hand Side Sensitivity Analysis**
The right hand side of the sensitivity analysis tells us how the changes in the right hand side make the changes in the optimality of the programme this is a case in the post optimal analysis the allowable decrease or increase in the constraints
We have three constraints c1 ,c2 and c3
The allowable decrease for the c1 is 340
The allowable decrease for the c2 is 720
The allowable decrease for the c3 is 240

The allowable increase for the constraints are given by

Allowable Increase
1E+30
1200
240

2.Consider the following problem:
$f(x1, x2) = 4x1 + 6x2 - 2x1\ 2 - 2x1x2 - 2x2\ 2$
a) Write a program to visualize the above function.

FINAL LAB TEST MDSC-103P

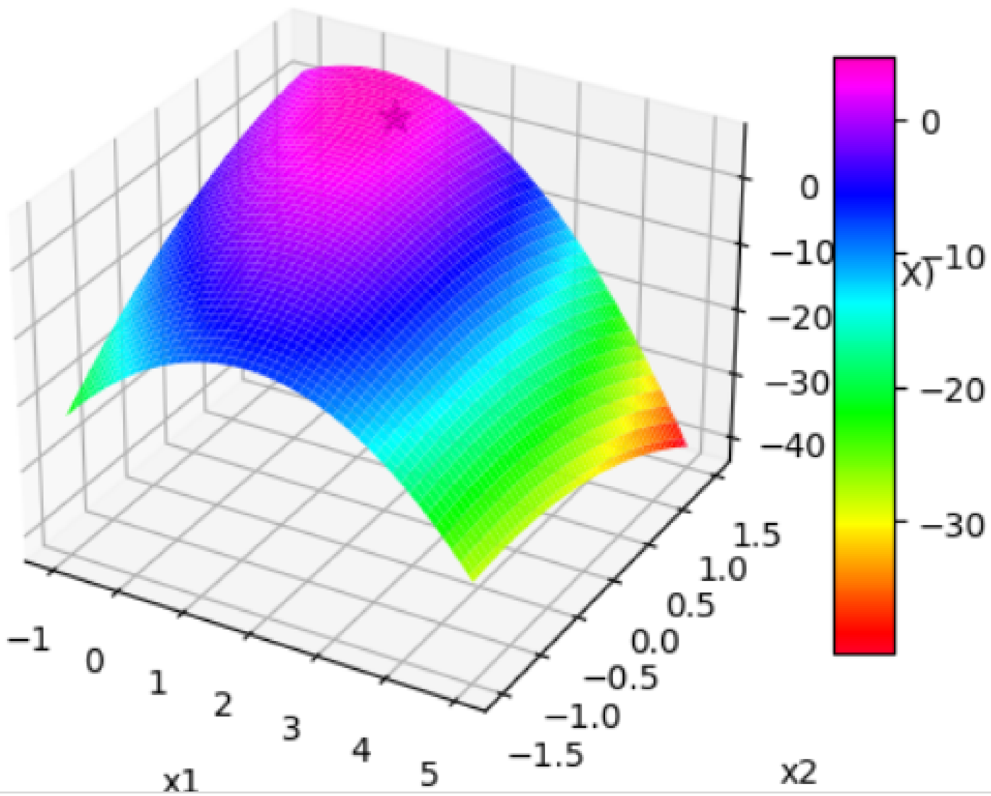
```
import numpy as np
import matplotlib.pyplot as plt
import scipy as sp
from scipy.special import gamma
```

�$(x1, x2) = 4x1 + 6x2 - 2x1^{**}2 - 2x1x2 - 2x2^{**}2$

```
[8] from mpl_toolkits.mplot3d import Axes3D
    f=lambda x1,x2:(4*x1+6*x2-2*x1**2-2*x1*x2-2*x2**2) # writing the function using lambda
    x2=np.linspace(-1.5,1.5,100) #x2 will take 100 equal data points ranging from 1.5 to 1.5
    x1=np.linspace(-1,5,100)# x1 will take the 100 equal data points from -1 to 5

    X,Y=np.meshgrid(x1,x2) #creating a mesh grid for the x1 and x2
    F=f(X,Y)
```

```
fig=plt.figure(figsize=(12,8)) #we are plotting the the figure size is of the ratio of 12 ,8
ax=plt.subplot(1,2,1,projection='3d')
ax.scatter3D(1,1,f(1,1),c="black",marker="*",s=100)
surf = ax.plot_surface(X,Y,F, cmap= 'gist_rainbow')
fig.colorbar(surf,shrink=0.4, aspect=10)
# set axes label
ax.set_xlabel('x1', labelpad=10)
ax.set_ylabel('x2', labelpad=30)
ax.set_zlabel('f(X)', labelpad=10)
plt.show()
```



FINAL LAB TEST MDSC-103P

b) Write an iterative program to maximize the function

```
f=lambda x1,x2:-(4*x0+6*x1-2*x0**2-2*x0*x1-2*x1**2)
x0=1
x1=5
delta=1*10**(-5)
eta=x0-x1
learning_rate=0.1

#grad function
def gradf(x):
  return -(4*x0+6*x1-2*x0**2-2*x0*x1-2*x1**2)

x1=x0-eta*gradf(x0)
x0=x1
a=[]

for i in range(0,100):
  x1=x0-eta*gradf(x0)
  x0=x1
  if gradf(x0)<delta:
    break
```

```
print(x1)
  #graph plot with x,f(x) values with the plot function
plt.scatter(a,f(np.array(a)))
plt.plot(x,f(x))
plt.title("plot for alll the x ,f(x) values")
plt.xlabel("x")
plt.ylabel("f(x)")
plt.show()
```