End Semester Examination

MDSC-103-(P)

Date: 10-10-23

1. **Formulate the problem in the Excel file and generate the sensitivity analysis.**
   ANSWER:

   Microsoft Excel 16.0 Sensitivity Report
   Worksheet: [23913_ese.xlsx]Sheet1
   Report Created: 10-10-2023 13:39:41

   Variable Cells

Cell	Name	Final Value	Reduced Cost	Objective Coefficient	Allowable Increase	Allowable Decrease
D15	cowhide	360	0	7	1.333333333	2
E15	production	300	0	10	4	1.6

   Constraints

Cell	Name	Final Value	Shadow Price	Constraint R.H. Side	Allowable Increase	Allowable Decrease
F17		3600	1	3600	1200	720
F18		960	2	960	240	240

2. **Write on cost coefficient sensitivity analysis.**
   ANSWER:

   Variable Cells

Cell	Name	Final Value	Reduced Cost	Objective Coefficient	Allowable Increase	Allowable Decrease
D15	cowhide	360	0	7	1.333333333	2
E15	production	300	0	10	4	1.6

   - From the above Picture and Sensitivity Report we can infer that baseballs and softballs final value is 360 and 300. Which means we are producing the 360 baseballs and 300 softballs and getting 5520 dollars as a profit and which is optimal solution

- Reduced cost is 0 for both, which indicates that if we increase the production of balls, that won't affect the profit.

- The maximum increase in dollars
  Baseballs prize is $8.333
  Softballs prize is $14
  (IF the prize goes more than that, it might lead to non-optimal solution)
- The maximum decrease in dollars
  Baseballs prize is $5
  Softballs prize is $8.4
  (IF the prize goes less than that, it might lead to non-optimal solution)

3. **Write on Right Hand Side Sensitivity Analysis**
   ANSWER:

Constraints

Cell	Name	Final Value	Shadow Price	Constraint R.H. Side	Allowable Increase	Allowable Decrease
F17		3600	1	3600	1200	720
F18		960	2	960	240	240

- Final value of cowhide covers and production is 3600 and 960. Utilization of the resources to the fullest.
- Shadow price of **cowhide is 1** and **production is 2**, which indicates that in future some modification is there in the system. We can put more time on production instead of cowhides cover because it contributes to profit double that time cowhides cover.
- The maximum increase
  Cowhide cover is 4800 square feet
  Production time is 1200 minutes
  (IF the production time and cowhide sheet goes more than that, it might lead to non-optimal solution)

- The maximum decrease

  Cowhide cover is 2880 square feet

  Softballs prize is 720 minutes

  (IF the production time and cowhide sheet goes more than that, it might lead to non-optimal solution)

---

2. Consider the following problem:

$$f(x_1, x_2) = 4x_1 + 6x_2 - 2x_1^2 - 2x_1x_2 - 2x_2^2$$

   a) Write a program to visualize the above function.
   b) Write an iterative program to maximize the function.

Consider the following problem:

f(x1+x2) = 4x1 + 6x2 - 2x1^2 - 2x1x2 - 2x2^2

Visualizing

```python
# importing the important libraries
import numpy as np #for using np.arange
import matplotlib.pyplot as plt  # for plotting
from scipy.optimize import minimize   #for minimizing problem
```

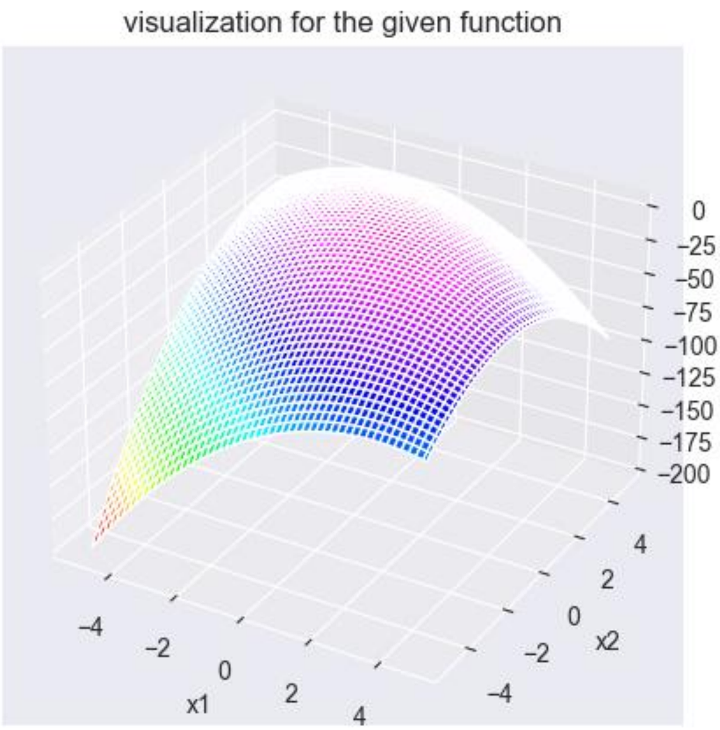
✓ 0.0s Python

```python
# defining te user defined funtion for ploting the function
def fun(x1,x2):
    ans = 4*x1 + 6*x2 - 2*(x1**2) - 2*x1*x2 - 2*(x2**2)
    return ans # returing the computed answer
```

✓ 0.0s

```python
x1 = np.arange(-5,5,.1)  #x1 values for -5 to 4.9 with different .1
x2 = np.arange(-5,5,.1)  #x2 values for -5 to 4.9 with different .1

X, Y = np.meshgrid(x1,x2)  # x1 and x2 will expanded by mesgrid function
F = fun(X,Y) # the F is calling function fun

fig = plt.figure(figsize=(10,10)) # figsize
ax = plt.subplot(1,1,1,projection='3d') # 3d plotting axis

surf = ax.plot_surface(X,Y,F, cmap= 'gist_rainbow') # setting the color to rainbow

ax.set_xlabel('x1') # label for x axis
ax.set_ylabel('x2') # label for y axis
ax.set_title("visualization for the given function") #title
ax.set_zlabel('f(x1,x2)') # label for z axis
```

✓ 1.8s



MAXIMIZING THE FUNCTION

```python
# function for f(x)
def f(x):
    x1, x2 = x
    return -4*x1 - 6*x2 + 2*(x1**2) + 2*x1*x2 + 2*(x2**2) # fucntion is mul with (-) to make to max


bounds = ((None, None), (None, None)) # bounds


initial_guess = [0, 0]  # initial vaules
result = minimize(f, initial_guess, method='SLSQP', bounds=bounds) #'using SLSQP we are maximiing the function'

print(result.x)
result
```

✓ 0.0s

```
[0.33333333 1.33333333]

 message: Optimization terminated successfully
 success: True
  status: 0
     fun: -4.666666666666668
       x: [ 3.333e-01  1.333e+00]
     nit: 3
     jac: [ 0.000e+00  0.000e+00]
    nfev: 11
    njev: 3
```