Linear Programing and Sensitivity Analysis in Python

Alireza Sheikh-Zadeh, Ph.D.

Sensitivity Analysis

Finding the optimal solution to a linear programming model is important, but it is not the only information available. There is a tremendous amount of sensitivity information, or information about what happens when data values are changed. Even though we may have solved a model to find an optimal solution, it would be beneficial to determine what impact a change in a price or cost would have on net profit. A main purpose of sensitivity analysis is to identify the sensitive parameters (i.e., those that cannot be changed without changing the optimal solution).

Some Terminology

Before diving into the sensitivity analysis, we define some relevant terminology.

Binding constraints: are the constraints that hold with equality at the optimal solution.

Slack: or Surplus is the difference between the two sides of each constraint at optimal solution.

Shadow price: tells how much the objective value will change if the right hand side (RHS) of a constraint is increased or decreased by 1 unit. Depending on the status of the constraint on the optimal solution:

- If the constraint is binding (zero slack), because the limited supply of resources binds the decision variable from being increased further, the *shadow prices* of the constraint (in the maximization problem) is positive. The shadow price denotes to the economical vale of the constraint's right hand side. Economists refer to such resources as scarce goods (resources with positive shadow price).

- If a constraint is not binding (non-zero slack), so resources are available in surplus (E.g., constraint 1 in the WYNDOR GLASS LP model), and the shadow price is zero. Economists refer to such resources as free goods (resources with a zero shadow price).

Reduced Cost: Sometimes the optimal solution for some variables is Zero. For example, suppose the optimal value of the variables for a maximization problem is $x_1^* = 0, x_2^* = 2$, this implies that variable x_1 is not profitable enough (in the maximization problem), so it stays zero. The **reduced cost** is the amount that the *objective coefficient of the variable* would have to be increased before it would become profitable to give the variable a positive value in the optimal solution. For example, if a variable (e.g., x_1) had a reduced cost of 10, the objective coefficient of that variable (e.g., if the objective function is $Z = 15x_1 + 20x_2$, the objective coefficient of x_1 is 15) would have to be increased by 10 units in a maximization problem (e.g., the objective changes to $Z = 25x_1 + 10x_2$) and/or decrease by 10 units in a minimization problem (e.g., the objective changes to $Z = 5x_1 + 10x_2$) for the variable to become an attractive alternative to enter into the solution.

Note: Typically, greater attention is given to performing sensitivity analysis on the RHS and objective function coefficients than on the constraints coefficients. On real problems with hundreds or thousands of constraints and variables, the effect of changing one constraint coefficient is usually negligible, but changing one RHS or objective function coefficient can have real impact.

Example: WYNDOR GLASS CO. Sensitivity Analysis

Max

$Z = 3x_1 + 5x_2$

S.t.

$x_1 \leq 4$

$2x_2 \leq 12$

$3x_1 + 2x_2 \leq 18$

$x_1, x_2 \geq 0$

```
from pulp import *


prob = LpProblem("WYNDOR GLASS CO.", LpMaximize)

x1 = LpVariable('x_1', lowBound = 0)
x2 = LpVariable('x_2', lowBound = 0)

# Objective function
prob += 3*x1 + 5*x2, "Profit"

# Constraints
prob += x1 <= 4, "C1: Plant 1"
prob += 2*x2 <= 12, "C2: Plant 2"
prob += 3*x1 + 2*x2 <= 18, "C3: Plant 3"

print(prob)
```

```
## WYNDOR GLASS CO.:
## MAXIMIZE
## 3*x_1 + 5*x_2 + 0
## SUBJECT TO
## C1:_Plant_1: x_1 <= 4
```

```
##
## C2:_Plant_2: 2 x_2 <= 12
##
## C3:_Plant_3: 3 x_1 + 2 x_2 <= 18
##
## VARIABLES
## x_1 Continuous
## x_2 Continuous
```

```
prob.solve()
```

```
## 1
```

```
print(LpStatus[prob.status])
```

```
## Optimal
```

```
for variable in prob.variables():
    print("{}* = {}".format(variable.name, variable.varValue))


print(value(prob.objective))
## 36.0
```

```
# We add these lines for sensitivity analysis
```

```
print("\n Sensitivity Analysis: ")
```

```
##
##  Sensitivity Analysis:
```

```
for name, c in prob.constraints.items():
    print("\n", name, ":", c, ", Slack=", c.slack, ", Shadow Price=", c.pi)
```

```
##
##  C1:_Plant_1 : x_1 <= 4 , Slack= 2.0 , Shadow Price= -0.0
##
##  C2:_Plant_2 : 2*x_2 <= 12 , Slack= -0.0 , Shadow Price= 1.5
##
##  C3:_Plant_3 : 3*x_1 + 2*x_2 <= 18 , Slack= -0.0 , Shadow Price= 1.0
```

```
for v in prob.variables():
  print ("\n", v.name, "=", v.varValue, ", Reduced Cost=", v.dj)
```

```
##
##  x_1 = 2.0 , Reduced Cost= 0.0
##
##  x_2 = 6.0 , Reduced Cost= 0.0
```

Since the objective is expressed in thousands of dollars of profit per week, Shadow-Price = 1.5 for constraint 2 indicates that adding 1 more hour of production time per week in Plant 2 for these two new products would increase their total profit by $1,500 per week. Should this actually be done? It depends on the marginal profitability of other products currently using this production time. If there is a current product that contributes less than $1,500 of weekly profit per hour of weekly production time in Plant 2, then some shift of production time to the new products would be worthwhile.

Some questions:

If the objective function coefficients changes, how does the solution change?

Answer: It changes the optimal objective value.

If the available resources (RHS) changes, how does the solution change?

Answer: Depending on the shadow price, it changes the optimal objective value.

If a constraint is added to the problem, how does the solution change?

Answer: It may reduce the size of the feasible region and affect the optimal objective value. For the maximization (minimization) problem it may reduce (increase) the optimal objective value.

One important caution: the sensitivity report information applies to changes in only one of the model parameters at a time; all others are assumed to remain at their original values. In other words, you can not accumulate or add the effects of sensitivity information if you change the values of multiple parameters in a model simultaneously.

Example: Cost minimization problem

Min

$Z = 3x_1 + 2x_2$

S.t.

$x_1 \geq 2$

$2x_1 + x_2 \geq 6$

$x_1, x_2 \geq 0$

```
# Model in Python
from pulp import *
# Define the Model
prob = LpProblem("Production Cost", LpMinimize)
x1 = LpVariable('x_1', lowBound = 0)
x2 = LpVariable('x_2', lowBound = 0)
# Objective function
prob += 3*x1 + 2*x2, "Obj"
# Constraints
prob += x1 >= 2
prob += 2*x1 + x2 >= 6
# print(prob)
```

```
prob.solve()

## 1

print("status: " + LpStatus[prob.status])

## status: Optimal

for variable in prob.variables():
     print("{}* = {}".format(variable.name, variable.varValue))


## x_1* = 3.0
## x_2* = 0.0

print(value(prob.objective))

## 9.0

print("\n Sensitivity Analysis: ")

##
##  Sensitivity Analysis:

for name, c in prob.constraints.items():
    print("\n", name, ":", c, ", Slack=", c.slack, ", Shadow Price=", c.pi)

##
##  _C1 : x_1 >= 2 , Slack= -1.0 , Shadow Price= 0.0
##
##  _C2 : 2*x_1 + x_2 >= 6 , Slack= -0.0 , Shadow Price= 1.5

for v in prob.variables():
  print ("\n", v.name, "=", v.varValue, ", Reduced Cost=", v.dj)

##
##  x_1 = 3.0 , Reduced Cost= 0.0
##
##  x_2 = 0.0 , Reduced Cost= 0.5
```

Real World Example: Portfolio Investment Risk Modeling

What is the best investment portfolio?

An investor has $500,000 to invest in the stock market and is considering investments in six stocks with the following current projected rates of return:

- Tech Stocks: Cisco (8%), Microsoft (6%) and Intel (5%),
- Bank Stocks: B of A (7%), First Bank (4%)
- Money Market: ING (2%)

The risk measure based on the historical volatility of the investments are:

- Tech Stocks: Cisco (14.02), Microsoft (10.57) and Intel (13.22),
- Bank Stocks: B of A (9.36), First Bank (7.61)
- Money Market: ING(2.39)

An investment company recommends that:

- At least as much in bank stocks as tech stocks
- No more than $200,000 be invested in any stock
- At least $50,000 be invested in ING

The investor would like to have an average return of at least 5% but would like to minimize the risk. What portfolio would achieve this?

Answer:

```
# Portfolio Investment Risk Modeling

from pulp import *

prob = LpProblem("My LP Problem", pulp.LpMinimize)

x1 = LpVariable('Cisco', lowBound = 0, upBound = 200)
x2 = LpVariable('Microsoft', lowBound = 0, upBound = 200)
x3 = LpVariable('Intel', lowBound = 0, upBound = 200)
x4 = LpVariable('BofA', lowBound = 0, upBound = 200)
x5 = LpVariable('FirstBank', lowBound = 0, upBound = 200)
x6 = LpVariable('ING', lowBound = 50, upBound = 200)

# Objective function
prob += 14.02*x1 + 10.57*x2 + 13.22*x3 + 9.36*x4 + 7.61*x5 + 2.39*x6, "Obj"
# Constraints
prob += x1 + x2 + x3 + x4 + x5 + x6 == 500, "Portfolio"
prob += 0.08*x1 +  0.06*x2 + 0.05*x3 + 0.07*x4 + 0.04*x5 + 0.02*x6 >=25, "Ret
urn"
prob += x4 + x5 >= x1+ x2 + x3, "Balance"

# print(prob)

prob.solve()

## 1

print("status: " + LpStatus[prob.status])

## status: Optimal

for variable in prob.variables():
    print("{}* = {}".format(variable.name, variable.varValue))
```

```
## BofA* = 200.0
## Cisco* = 50.0
## FirstBank* = 0.0
## ING* = 200.0
## Intel* = 0.0
## Microsoft* = 50.0
```

```
print(value(prob.objective))
```

```
## 3579.5
```

```
print("\n Sensitivity Analysis:")
```

```
##
##  Sensitivity Analysis:
```

```
for name, c in prob.constraints.items():
    print("\n{}: Slack={}, Shadow Price = {}".format(name, c.slack, c.pi)) #
OR
#    print("\n", name, ":", ", Slack=", c.slack, ", Shadow Price=", c.pi)
```

```
##
## Portfolio: Slack=-0.0, Shadow Price = 0.22
##
## Return: Slack=-0.0, Shadow Price = 172.5
##
## Balance: Slack=-100.0, Shadow Price = 0.0
```

```
print("\n ========================================")
```

```
##
##  ========================================
```

```
for variable in prob.variables():
     print("\n{}* = {},   Reduced Cost = {}".format(variable.name, variable.v
arValue, variable.dj))
```

```
##
## BofA* = 200.0,   Reduced Cost = -2.935
##
## Cisco* = 50.0,   Reduced Cost = 0.0
##
## FirstBank* = 0.0,   Reduced Cost = 0.49
##
## ING* = 200.0,   Reduced Cost = -1.28
##
## Intel* = 0.0,   Reduced Cost = 4.375
##
## Microsoft* = 50.0,   Reduced Cost = 0.0
```

Real World Example: Transportation problem

General Appliance Corporation (GAC) produces refrigerators at two plants: Marietta and Minneapolis. They ship them to major distribution centers in Cleveland, Baltimore, Chicago, and Phoenix.

The following Table shows the unit cost of shipping between any plant and distribution center, plant capacities over the next planning period, and distribution center demands.

Plant	Cleveland	Baltimore	Chicago	Phoenix	Capacity
Marietta	$12.60	$14.35	$11.52	$17.58	1,500
Minneapolis	$9.75	$16.26	$8.11	$17.92	800
Demand	150	350	500	1,000	

GAC's supply chain manager faces the problem of determining how much to ship between each plant and distribution center to minimize the total transportation cost, not exceed available capacity, and meet customer demand.