GUROBI OPTIMIZATION

Sensitivity analysis of LP problems

Gurobi Python API

Furniture Problem: economic interpretation

Economic interpretation in Linear Programming models

- Solving LP problems provides more information than only the values of the decision variables and the value of the objective function.

- Associated with an LP optimal solution there are ***shadow prices*** (a.k.a. ***dual variables***, or ***marginal values***) for the constraints.

- The shadow price of a constraint associated with the optimal solution, represents the change in the value of the objective function per unit of increase in the right-hand side value of that constraint.

- There are shadow prices associated with the non-negativity constraints. These shadow prices are called the ***reduced costs***.

Furniture Problem: economic interpretation .. 2

- For example, suppose the labor capacity is increased from 450 hours to 451 hours. What is the increase in the objective function value from such increase?
- Since the constraints on mahogany capacity (2.0) and labor capacity (3.0) define the optimal solution, we can solve the following system of equations

$$5x_1 + 20x_2 = 400 \quad \text{Mahogany capacity}$$
$$10x_1 + 15x_2 = 451 \quad \text{Labor capacity}$$

- The new values of the decision variables are: chairs (x1) = 24.16, tables (x2) = 13.96
- The new value of the objective function (revenue) is = $2,204
- The shadow price associated with the labor capacity is $2,204 - $2,200 = $4. That is, we can get $4 of increased revenue per hour of increase in labor capacity.
- **Remark:** The shadow price value of $4 remains constant over a range of value changes of the mahogany capacity. The calculation of this range is beyond the scope of this course.

Furniture Problem: economic interpretation .. 3

- Similarly, we can compute the shadow price of the mahogany constraint by solving the following system of equations

$$5x_1 + 20x_2 = 401$$ Mahogany capacity

$$10x_1 + 15x_2 = 450$$ Labor capacity

- The new values of the decision variables are: chairs (x1) = 14.08, tables (x2) = 23.88
- The new value of the objective function (revenue) is = $2,201
- The shadow price associated with the mahogany capacity is $2,201 - $2,200 = $1. That is, we can get $1 of increased revenue per unit of increase in mahogany capacity.
- **Remark:** The shadow price value of $1 remains constant over a range of value changes of the labor capacity.

Furniture Problem: simplex method revisited

The LP problem in a canonical form with respect to the optimal basic variables (x1, x2) is

$$Max \quad z = 2200 + 0x_1 + 0x_2 - 1h_1 - 4h_2$$

$$(2.0) \quad x_2 = 14 - \left({}^{2}/_{25}\right)h_1 + \left({}^{4}/_{25}\right)h_2$$

$$(3.0) \quad x_1 = 24 + (3/25)h_1 - (4/25)h_2$$

$$x_1, x_2, h_1, h_2 \geq 0$$

- **Remarks**: Recall that h1 and h2 are the slack variables associated with the mahogany capacity constraint and labor capacity constraint, respectively.
- The interpretation of the slack variables is the amount of resource capacity not consumed by the production of chairs and tables.
- The optimal solution is
  - Revenue = $2,200, chairs (x1) = 24, tables (x2)= 14
  - Slack variable (h1) of mahogany constraint = 0, and
  - Slack variable (h2) of labor constraint = 0

- **Remarks**: If we increase the value of h1 (unused capacity of the mahogany) by one the total revenue will be reduced by $1. If we increase the value of h2 (unused capacity of labor) by one the total revenue will be reduced by $4.
- Our previous analysis show us that the shadow price of mahogany is $1 and the one of labor is $4 !!!!
- **Conclusion**: The simplex method automatically give us the shadow prices of the resources.

Furniture Problem: solved with Gurobi

This method adds constraints to the model object f. We store the constraints generated in an object called (res).

```
res = f.addConstrs(((sum(bom[r,p]*make[p] for p in products) <= capacity[r]) for r in resources), name='R')
```

For each resource constraint in the dictionary (res), check if its associated shadow price is greater than zero. Then print the resource constraint name and the resource constraint shadow price.

Recall that the object (res) stores all the information related to the constraints of the model f.

```
# display shadow prices of resources constraints
for r in res:
	if (abs(res[r].Pi) > 1e-6):
		print(res[r].ConstrName, res[r].Pi)
```

```
('R[mahogany]', 1.0)
('R[labor]', 4.0)
```

Furniture Problem: economic interpretation .. 4

- **Is it profitable to make a third product, like desks?**
  - **Assume that the price of the desk is $110,**
  - **and the desk consumes 15 units of mahogany and 25 units of labor**

    - The previous python code has parametrized the Furniture LP model, i.e. the model formulation does not depend on the data of the problem. Therefore, we just generate a new set of data that includes the new product information

```
# products data,
products, price  = multidict({
    'chair': 45,
    'table':  80,
    'desk': 110 })
```

```
# Bill of materials: resources required by each product
bom = {
    ('mahogany', 'chair'): 5,
    ('mahogany', 'table'):  20,
    ('mahogany', 'desk'): 15,
    ('labor', 'chair'):     10,
    ('labor', 'table'):   15,
    ('labor', 'desk'):   25 }
```

Furniture Problem: economic interpretation .. 5

- The new LP model is

```
# save model for inspection
f.write('furniture.lp')
```

```
\ Model Furniture
\ LP format - for model browsing. Use MPS format to capture full model detail.
Maximize
  80 make[table] + 45 make[chair] + 110 make[desk]
Subject To
 mahogany: 20 make[table] + 5 make[chair] + 15 make[desk] <= 400
 labor: 15 make[table] + 10 make[chair] + 25 make[desk] <= 450
Bounds
End
```

GUROBI OPTIMIZATION

Furniture Problem: economic interpretation .. 6

```
# run optimization engine
f.optimize()
```

```
# display optimal values of decision variables
for v in f.getVars():
    if (abs(v.x) > 1e-6):
        print(v.varName, v.x)

# display optimal total profit value
print('total profits', f.objVal)
```

```
('make[table]', 14.0)
('make[chair]', 24.0)
('total profits', 2200.0)
```

It is not profitable to produce desks. The optimal solution remains the same.

```
# display shadow prices of resources constraints
for r in res:
    if (abs(res[r].Pi) > 1e-6):
        print(res[r].ConstrName, res[r].Pi)
```

```
('R[mahogany]', 1.0)
('R[labor]', 4.0)
```

The shadow prices of the resources remain the same.

Furniture Problem: economic interpretation .. 7

- Notice that we can use the shadow price information of the resources to check if it is worth it to make desks.
  - The shadow price of the mahogany capacity constraint is $1
  - The shadow price of the labor capacity constraint is $4
  - Let’s compute the opportunity cost of making one desk and compare it with the price of a desk. If this opportunity cost is greater than the price, then it is not worth it to make desks.
  - The opportunity cost can be computed by multiplying the units of mahogany capacity that one desk built consumes by the shadow price of mahogany capacity, and multiplying the hours of labor capacity that one desk built consumes by the shadow price of labor capacity:
    - That is, ($1)*15 (units of mahogany) + ($4)*25 (hours of labor) = $115 > $110
- Therefore, investing resources to produce desks, otherwise used to produce chairs and tables, is not profitable.