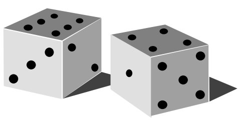
BAN 630
Balaraman Rajan

balaraman.rajan@csueastbay.edu

Non-linear optimization models



Learning Objectives

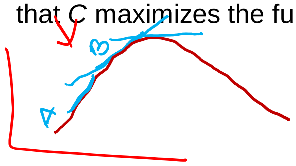
- Define nonlinear programming (NLP) modeling and optimization
- Describe and interpret local maxima and global maximum of NLP problems
- Use steps of solving NLP optimization model
- Formulate and solve, using NLP models, a range of real applications
- Apply Python and Gurobi to identify optimal solution for these applications using NLP modeling
- Analyze sensitivity of optimal solutions

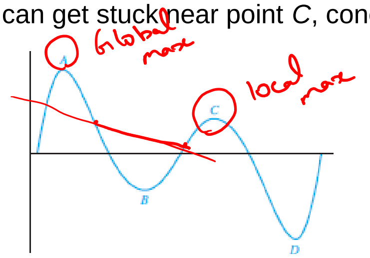
Why Nonlinear Programming (NLP) Models?

- A model can become nonlinear for several reasons, including the following:
  - There are non-constant returns to scale (for example, price or profit per unit), which means that the effect of some input on some output is nonlinear
    - In pricing models, price *p* is a variable, and quantity of units sold *D* could be a function of price, $\boxed{D = f(p)}$. Thus, maximizing revenue *R*, as a product of price and demand $R = D*p = f(p)*p$, make the objective function nonlinear
    - In financial models, the risk can be measured as the variance (or standard deviation) of the portfolio, which is a nonlinear function of the decision variables. Thus, a financial model that maximizes return and minimizes risk becomes a nonlinear model
- The real world *often behaves in a nonlinear manner*, so when you model a problem with LP, you are typically approximating reality
- By allowing nonlinearities in your models, you can often create more realistic models. Unfortunately, this comes at a price - nonlinear optimization models *are more difficult to solve*

Basic Ideas of NLP Optimization

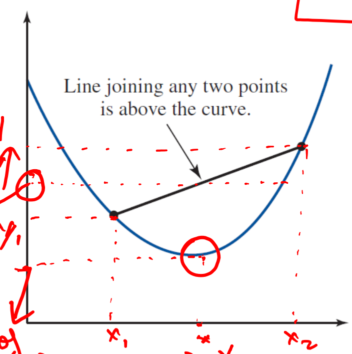
- When you solve an LP problem, you are guaranteed that the algorithm solution, if it exists, is optimal
- When you solve an *NLP problem*, however, the algorithm sometimes obtains a suboptimal solution
- For the figure graphed below, points *A* and *C* are called *local maxima* because the function is larger at *A* and *C* than at nearby points
- However, only point *A* maximizes the function; it is called the *global maximum*
- The problem is that algorithms can get stuck near point *C*, concluding that *C* maximizes the function





Convex and concave functions (1 of 3)

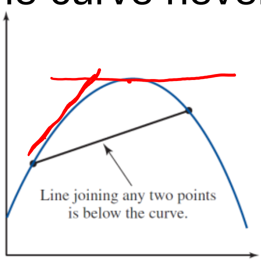
- Algorithms are guaranteed to solve convex and concave functions.
- A function of one variable is ***convex*** in a region if
  - its slope (rate of change) in that region is always non-decreasing.
  - a line drawn connecting two points on the curve never lies below the curve.

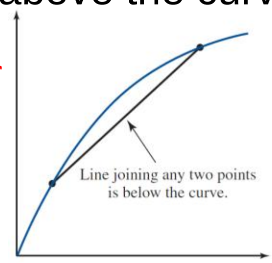


$\text{Avg}(y_1, y_2) > y(\text{Avg of } x_1, x_2)$

Convex and concave functions (2 of 3)

- In contrast, the function is concave if its slope is always nonincreasing, or equivalently, if a line connecting two points on the curve never lies above the curve.





Max (Profit) = Max [Rev − Cost]

Convex and concave functions (3 of 3)

- It can be shown that the sum of convex functions is convex and the sum of concave functions is concave.

- If you multiply any convex function by a positive constant, the result is still convex, and if you multiply any concave function by a positive constant, the result is still concave.

- However, if you multiply a convex function by a negative constant, the result is concave, and if you multiply a concave function by a negative constant, the result is convex.

How to Solve for NLP

- Main approach in optimizing NLP model is *Generalized Reduced Gradient (GRG)* method
  - Process of solving an optimization problem where the objective function and/or some (or all) of the problem constraints are nonlinear
  - An optimization problem is one of calculation of the extrema (maxima or minima) of an objective function over a set of unknown real variables with the condition of satisfying problem constraints
- Because there is then some doubt whether the solution is the optimal solution, the best strategy is to
  1. Try several possible starting values for the changing cells
  2. Run the algorithm from each of these
  3. Take the best solution
- In general, if you try several starting combinations for the changing cells and the algorithm obtains the same optimal solution in all cases, you can be fairly confident - but still not absolutely sure - that you have found the optimal solution to the NLP
- On the other hand, if you try different starting values for the changing cells and obtain several different solutions, then all you can do is to keep the best solution you have found and hope that it is indeed optimal

Multistart Option for NLP

- Because it is difficult to know where to start, the *Multistart* feature provides an automatic way of starting from a number of starting solutions
- It selects several starting solutions automatically, runs the GRG nonlinear algorithm from each, and reports the best solution it finds
- https://support.gurobi.com/hc/en-us/articles/360043834831-How-do-I-use-MIP-starts

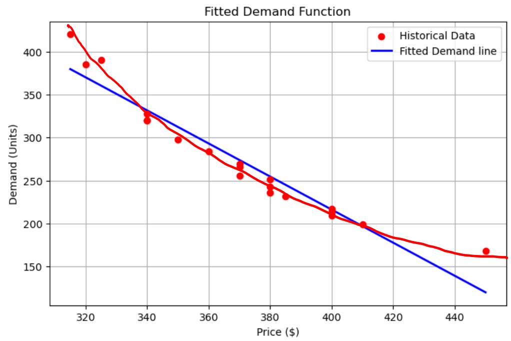
Pricing Suits at Sullivan's

- Sullivan's is a retailer of upscale men's clothing. Suits cost Sullivan's $320 per unit. The historical data of Sullivan's prices and associated suit demands is summarized in the table on the right.
- Each purchase of a suit leads to an average of 2 shirts and 1.5 ties being sold with this suit. Each shirt contributes $25 to the profit, and each tie contributes $15 to the profit.
- The dollar figures are supplied by a cost accountant. The demand function can be estimated using the historical data of the relationships between demand and price. The average numbers of shirts and ties sold with suit purchases are also available from historical data. Sullivan's would like to have a suit's price to be at least $320.
- Use an NLP model to determine a profit-maximizing price for suits, taking into account the purchases of shirts and ties that typically accompany purchases of suits.

Period	*Price*	*Demand*
1	$360.00	284
2	$340.00	320
3	$380.00	251
4	$410.00	199
5	$320.00	385
6	$350.00	298
7	$400.00	217
8	$370.00	256
9	$340.00	320
10	$450.00	168
11	$400.00	214
12	$380.00	243
13	$370.00	269
14	$315.00	420
15	$340.00	328
16	$325.00	390
17	$370.00	266
18	$380.00	236
19	$400.00	209
20	$385.00	232

Modeling Demand Function for Sullivan's Suits

Period	Price	Demand
1	$360.00	284
2	$340.00	320
3	$380.00	251
4	$410.00	199
5	$320.00	385
6	$350.00	298
7	$400.00	217
8	$370.00	256
9	$340.00	320
10	$450.00	168
11	$400.00	214
12	$380.00	243
13	$370.00	269
14	$315.00	420
15	$340.00	328
16	$325.00	390
17	$370.00	266
18	$380.00	236
19	$400.00	209
20	$385.00	232



Sullivan's Suits: Model Formulation

Decision variable:

p = price per suite, $

Inputs:

D = demand based on price, units

D = *985.23 - 1.92 * p*

c = cost per suit, *$320*

s = Profit contribution per shirt, *$25*

t = Profit contribution per tie, *$15*

Objective:

Price Cost

Maximize profit: *max [(p - c)D + 2sD + 1.5tD]*

(p-c) (985.23 -1.92p) + 2 - - - .

Constraints:

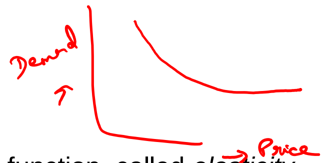
Minimum price: *p >= 320*

Non-negativity: *p >= 0*

Demand Elasticity Function for Sullivan Suits

- Common function for the price-dependent demand is a *constant elasticity demand function*:

$$D = ap^b$$



- In case of Sullivan's Suits, it is

$$D = 2{,}679{,}572{,}856\ p^{-2.73}$$

- Parameter *b* of the constant elasticity demand function, called *elasticity of demand*, is the percentage of change in demand caused by 1% of variation in price
  - If $b = -2.62$, demand decreases by *2.62%* when prices increases by *1%*
  - If prices decreases by *1%,* demand increases by *2.62%*
- Parameter *a* of the constant elasticity demand function describes the value of demand when the price is equal to *1*

Pickens Memorial Hospital

- The weekly income at Pickens Memorial Hospital depends on the number of patients admitted in the three separate categories: medical, surgical, and pediatric. The hospital can admit a total of 100 medical patients, 80 surgical patients, and 60 pediatric patients each week. However, because Pickens Memorial serves a large community, patient demand in each category by itself exceeds the total patient capacity.
- Due to fixed overhead, the income per patient in each category actually increases as the number of patients increases. Further, some patients who are initially classified as medical patients then get reclassified as surgical patients. As a result, the income per surgical patient also depends on the number of medical patients admitted. The accountants at Pickens Memorial have analyzed the situation, and have identified the following information:
  - Income contribution per medical patient: *= \$120 + \$8M*
  - Income contribution per surgical patient: *= \$200 + \$10S + \$3M*
  - Income contribution per pediatric patient: *= \$95 + \$9P*
  - Where M = number of medical patients admitted; S = number of surgical patients admitted; and P = number of pediatric patients admitted

$(120 + 8M) \cdot M$

$(200 + 10S + 3M) \cdot S$

$(95 + 9P) \cdot P$

- Perkins Memorial identified three main constraints for this model: scanning capacity (x-rays, MRI, and CT scans), surgical rooms capacity, and lab capacity. Table below shows the relevant weekly data for these three constraints for each category of patient. The table also shows the weekly availabilities of each of these resources: total number of scanning and surgical procedures is 420 and 90, respectively, and total hours of testing is 1150 hours.

Decisions	*Number of scannings per Patient*	*Number of Surgical Procedures per Patient*	*Number of Lab Tests per Patient*
Medical patients	1		2
Surgical patients	3	1.15	2.5
Pediatric patients	2		2
Total resource availability	420	90	1150

$M + 3S + 2P \le 420$

$1.15S \le 90$

$(2M + 2.5S + 2P)\,(2 + 0.001(M+S+P)) \le 1150$

- The hospital's chief laboratory supervisor has noted that the required time per lab test increases as the total number of patients increases. Based on historical data, the supervisor estimates this relationship to be as follows:
  - Time required per lab test (in hours) *= 2 + 0.001(M + S + P)*

(Effect of Congestion)

- Develop and solve an NLP model that identifies the optimal number of patients to be admitted to the Pickens Memorial Hospital that maximizes the total weekly income in the hospital.

Formulation of NLP Model for Pickens Memorial Hospital

Decision Variables:

M = number of medical patients admitted
S = number of surgical patients admitted
P = number of pediatric patients admitted

Objective: maximize gross income

Max $(\$120 + \$8M)M + (\$200 + \$10S + \$3M)S + (\$95 + \$9P)P$ → Non-linear

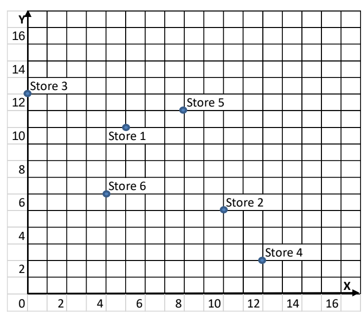
Constraints:

Capacity for medical patients:	$M \leq 100$	Linear constraints
Capacity for surgical patients:	$S \leq 80$	
Capacity for pediatric patients:	$P \leq 60$	
Scanning capacity:	$M + 3S + 2P \leq 420$	
Surgical capacity:	$1.15S \leq 90$	
Lab capacity, hours:	$(2M + 2.5S + 2P)(2 + 0.001(M + S + P)) \leq 1150$	Non-linear Constraint
Nonnegativity and integer	$M, S, P \geq 0$ *and integer*	

W-Mart Warehouse Location

- W-Mart Co., a retail company, would like to build a major distribution warehouse in the San Francisco Bay Area that will distribute/ship items to 6 local W-Mart stores. The location of the stores in terms of their graphical *x, y* coordinates, measured in miles relative to the point *x = 0, y = 0*, and the expected number of annual shipments from the warehouse to each store are presented in the following table:

Retail Stores	X-coord., miles	Y-coord., miles	Number of Annual Shipments	
Store 1	5	10	180	1
Store 2	10	5	300	1
Store 3	0	12	200	1
Store 4	12	2	400	1
Store 5	8	11	290	1
Store 6	4	6	160	100



- Identify the location (coordinates) of the distribution warehouse in order to minimize the total annual shipping distances from warehouse to the stores.

Warehouse Location: Model Formulation

$|(x_w - x_i)| + |(y_w - y_i)|$
Manhattan distance

Inputs:

i – Store index, $i = 1, \ldots, 6$

x_i, y_i = Coordinates of store i

s_i = Projected number of annual shipments from warehouse to store i

?

Decision variables:

x_w, y_w = Coordinates of warehouse

Calculated value:

d_i = Distance between warehouse and store i:

Euclidean distance

$$d_i = \sqrt{(x_w - x_i)^2 + (y_w - y_i)^2}$$

Objective: minimize total annual shipping distances:

Min

$$\sum_{i=1}^{6} s_i d_i$$

Model Formulation with Warehouse Location Constraints

Inputs:

i – Store index, $i = 1, \ldots, 6$

x_i, y_i = Coordinates of store i

s_i = Projected number of annual shipments from warehouse to store I

X_L, X_U = Lower and upper x-coordinate limits for warehouse location ($6,14$)

Y_L, Y_U = Lower and upper y-coordinate limits for warehouse location ($8,12$)

Decision variables:

x_w, y_w = Coordinates of warehouse

Calculated value:

d_i = Distance between warehouse and store i:

$$d_i = \sqrt{(x_w - x_i)^2 + (y_w - y_i)^2}$$

Objective: minimize total annual shipping distances:

Min:

$$\sum_{i=1}^{6} s_i d_i$$

Constraints:

X-coordinate limits for warehouse location:

$$x_w \le X_U$$
$$x_w \ge X_L$$

Y-coordinate limits for warehouse location:

$$y_w \le Y_U$$
$$y_w \ge Y_L$$