Assignment 2

Exercise 1

This problem is to use sensitivity analysis on revenue management for a very simplified airline pricing model. We will assume that an airline has one flight per day from Boston to Atlanta and they use an airplane that seats 150 people. The airline wishes to determine three prices, p_i (i=1,2,3), one for seats in each of the three fare buckets it will use. The fare buckets are designed to maximize revenue by separating travelers into groups, for instance 14 days advance purchase, leisure travelers, and business travelers. The airline models demand for seats in each group using the formula:

$$D_i = a_i \exp\left(-\frac{1}{a_i} p_i\right).$$

Where D_i is the people that want to fly given price p_i, the remaining parameters are a_1=100, a_2=150, and a_3=300. Please note, for simplicity you may assume that each D_i is a continuous variable.

(a) Formulate the revenue maximization problem for this flight as an optimization problem.
(b) What are the optimal prices and how many people are expected to buy a ticket in each fare bucket?
(c) Using sensitivity analysis, if the airline were to squeeze three additional seats onto this flight,
   a. How much do you expect revenue to change?
   b. By how much should the airline change each price?

Exercise 2

Consider the following three optimization problems:

The Banana (Rosenbrock) Function

This function is known as the "banana function" because of its shape; it is described mathematically in Equation (1). In this problem, there are two design variables with lower and upper limits of [-5, 5]. The Rosenbrock function has a known global minimum at [1, 1] with an optimal function value of zero.

$$\text{Minimize } f(\mathbf{x}) = 100\left(x_2 - x_1^2\right)^2 + \left(1 - x_1\right)^2 \qquad (1)$$

The Eggcrate Function

This function is described mathematically in Equation (2). In this problem, there are two design variables with lower and upper bounds of [-2π, 2π]. The Eggcrate function has a known global minimum at [0, 0] with an optimal function value of zero.

$$\text{Minimize } f(\mathbf{x}) = x_1^2 + x_2^2 + 25\left(\sin^2 x_1 + \sin^2 x_2\right) \qquad (2)$$

Golinski's Speed Reducer

This hypothetical problem represents the design of a simple gearbox such as might be used in a light airplane between the engine and propeller to allow each to rotate at its most efficient speed.

The gearbox is depicted in Figure 2 and its seven design variables are labeled. The objective is to minimize the speed reducer's weight while satisfying the 11 constraints imposed by gear and shaft design practices. A full problem description can be found in Reference [1]. A known feasible solution obtained by a sequential quadratic programming (SQP) approach is a 2994.34 kg gearbox with the following values for the seven design variables: [3.5000 0.7000 17.0000 7.3000 7.7153 3.3502 5.2867].

This is a feasible solution with four active constraints, but is it an optimal solution?

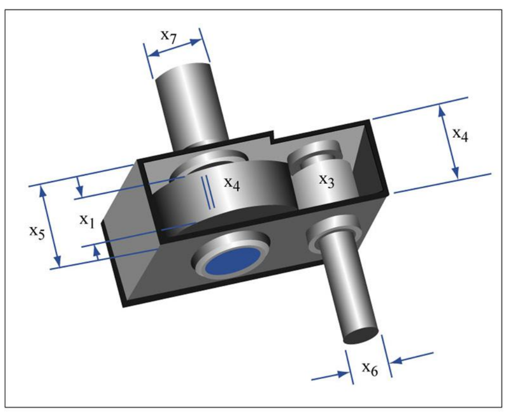

Image by MIT OpenCourseWare.

Figure 2**:** Golinski's Speed Reducer with 7 design variables

[1] Ray, T., "Golinski's Speed Reducer Problem Revisited," *the AIAA Journal*, Vol. 41, No. 3, 2003, pp. 556 -558.

Numerically find the minimum (=optimal) feasible design vector x for each of the above three problems using a gradient search technique of your choice. For each run record the starting point you used, the iteration history (objective value on y-axis and iteration number on x-axis), the final point at which the algorithm terminated and whether or not the final solution is feasible. Do at least 10 runs for each problem, but no more than 100.

Discuss the results and insights you get from numerically solving these three nonlinear optimization problems.

Exercise 3

Repeat the numerical experiments from exercise 2, but this time using a heuristic technique of your
choice (e.g. SA, GA …). Explain how you "tuned" the heuristic algorithm. Both SA and GA.

Compare your two algorithms (the gradient- search one and the heuristic one) from above quantitatively and qualitatively for the three problems as follows:

i. Dependence of answers on initial design vector (start point, initial population)
ii. Computational effort (CPU time [sec] or FLOPS)
iii. Convergence history
iv. Frequency at which the technique gets trapped in a local maximum

In order to answer this question, you *<u>do not need</u>* to implement your algorithms in some way *BUT* you MUST explain what algorithm is being used. Describe not just your conclusions, but also the process you followed. Do you think your conclusions would still apply for larger, more complex design optimization problems?