MAE 5930 Optimization
Homework 7 and 8

Purpose: The problems assigned help develop your ability to • solve optimization problems using optimality conditions. • formulate and solve larger problems in MATLAB. NOTE: Please write or type your formulations clearly so that a reader can understand what you are doing. You are welcome to use the equivalent functions in Python.

Problem 1: Solve the following problem analytically and numerically using `fmincon`.

$$\begin{aligned} \text{minimize} \quad & x_1^2 + x_2^2 \\ \text{subj. to} \quad & x_2 + x_1 - 2 \le 0 \\ & x_1^2 - x_2 - 4 \le 0 \end{aligned}$$

Problem 2: Solve the following problem analytically and numerically using `fmincon`.

$$\begin{aligned} \text{minimize} \quad & x_1^2 + x_2^2 \\ \text{subj. to} \quad & x_1 - 10 \le 0 \\ & x_1 - x_2^2 - 4 \ge 0 \end{aligned}$$

Problem 3: Solve the following problem analytically and numerically using `fmincon`.

$$\begin{aligned} \text{minimize} \quad & x_1^2 + x_2^2 \\ \text{subj. to} \quad & 4 - x_1 - x_2^2 \le 0 \\ & 3x_2 - x_1 \le 0 \\ & -3x_2 - x_1 \le 0 \end{aligned}$$

Problem 4: Solve the following problem analytically and numerically using `fmincon`.

$$\begin{aligned} \text{minimize} \quad & x_1 x_2 \\ \text{subj. to} \quad & x_1 + x_2 \ge 2 \\ & x_2 \ge x_1 \end{aligned}$$

Problem 5: Solve the following problem analytically and numerically using `fmincon`.

$$\begin{aligned} \text{minimize} \quad & -x_1 \\ \text{subj. to} \quad & -x_1 \le 0 \\ & -x_2 \le 0 \\ & x_2 + (x_1 - 1)^3 \le 0 \end{aligned}$$

Problem 6: Solve the following problem analytically and numerically using `fmincon`.

$$\begin{array}{ll} \text{minimize} & 2x_1^2 - x_2^2 \\ \text{subj. to} & x_1^2 x_2 - x_2^3 = 0 \end{array}$$

Problem 7: Solve the traveling salesman problem using MATLAB's genetic algorithm solver `ga`. Use the MTZ formulation below.

$$\begin{array}{ll} \text{minimize} & \sum_{i,j} d_{ij} x_{ij} \\ \text{subj. to} & \sum_i x_{ij} = 1 \quad \forall j \\ & \sum_j x_{ij} = 1 \quad \forall i \\ & u_1 = 1 \\ & 2 \leq u_i \leq n \quad \forall i \neq 1 \\ & u_i - u_j + 1 \leq (n-1)(1 - x_{ij}) \quad \forall i \neq 1, \forall j \neq 1 \\ & x_{ij} \in \{0, 1\} \quad \text{and} \quad u_i \in \mathbb{R} \end{array}$$

Generate random x and y locations for the cities using random numbers as

```
>> x = 20*rand(1,n);
>> y = 20*rand(1,n);
```

See how large you can make `n` and still solve the problem. Recall that you could solve about 40 cities using the integer programming approach.

Problem 8: The powered descent phase of a planar planetary landing is commonly posed as the following optimization problem.

$$\begin{array}{lll} \text{minimize} & \sum_i ||u_i|| & \\ \text{subj. to} & x_{i+1} = Ax_i + Bu_i + g & \forall i = 1, \ldots, N \\ & 0 < \rho_1 \leq ||u_i|| \leq \rho_2 & \forall i = 1, \ldots, N \\ & x_1 = a, \quad x_{N+1} = b & \end{array}$$

The decision variables are $x_i \in \mathbb{R}^4$ ($i = 1, \ldots, N+1$) and $u_i \in \mathbb{R}^2$ ($i = 1, \ldots, N$). The x's represent the lander's state vector: horizontal range, vertical altitude, horizontal velocity, and vertical velocity. The u's represent the lander's control vector coming from the thruster in the horizontal and vertical directions.

The objective minimizes the amount of fuel consumed in the descent. The discrete difference equation approximates the differential equations describing the dynamics. There are no nonlinear terms (such as drag) because the thrust force dominates all forces except gravity (denoted by g). The control constraint limits the thrust force between lower and upper bounds, i.e., the thruster cannot be turned off and it cannot provide infinitely large thrusts. Lastly, the lander is starting at a point $a \in \mathbb{R}^4$ and landing at a point on the surface $b \in \mathbb{R}^4$. Problem data is on the next page.

Solve the problem using `fmincon`, `ga`, and `patternsearch`. Plot the state and control trajectories as a function of time.

```
A = [1,0,0.1303,0; 0,1,0,0.1303; 0,0,1,0; 0,0,0,1];
B = [0.0085,0; 0,0.0085; 0.1303,0; 0,0.1303];
g = [0;-0.0832;0;-1.2779];
N = 499;
a = [1000;1500;-25;0];
b = [0;0;0;0];
rho1 = 4;
rho2 = 12;
```

If you would like to learn how to solve this problem in less than a second, sign up for Optimal Spacecraft Guidance in the spring!