UCK 337
Introduction to Optimization
Spring 2019-2020
Problem Set III

Asst. Prof. N. Kemal Ure
Department of Aeronautics
Istanbul Technical University

Due date is **June 10th.**

Homework Policy:

- All homework should be submitted through Ninova. No hard copy submission are needed/allowed.
- Late homeworks can be submitted by email with a -30 points penalty per day. (Hence after 4 late days you will not gain any scores by submitting your homework)
- No hand written solutions are allowed. If you type the solution in LaTeX, you will get $+10$ points. There are no extra points for solutions written in MS Word or any other word processing tool.
- There is a bonus problem at the end, which is worth extra $+10$ points. The bonus problem is significantly more difficult than the rest of the PSet. Only attempt at doing it after you have mastered the subject and solved all the other problems.
- The full score for this PSet is 100 points. However, you can get a total of 120 points if you type the solutions in LaTeXand solve all the problems correctly along with the bonus problem.
- You can discuss the solution strategies with your classmates, however you must completely type your solutions on your own. If any cheating is detected, you will get a -100 (yes that is a negative score) from the PSet.

Solving Linear Equations

- **Problem 1 (10 points):** Let $\boldsymbol{A} \in \mathbb{R}^{m\times n}$, $\boldsymbol{b} \in \mathbb{R}^m$, $rank\boldsymbol{A} = m$ and $\boldsymbol{x}_0 \in \mathbb{R}^n$. Consider the following optimization problem:

$$\begin{aligned} \text{minimize} \quad & \|\boldsymbol{x} - \boldsymbol{x}_0\| \\ \text{subject to} \quad & \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}. \end{aligned}$$

  Show that this problem has a unique solution given by:

$$\boldsymbol{x}^* = \boldsymbol{A}^T(\boldsymbol{A}\boldsymbol{A}^T)^{-1}\boldsymbol{b} + (\boldsymbol{I}_n - \boldsymbol{A}^T(\boldsymbol{A}\boldsymbol{A}^T)^{-1}\boldsymbol{A})\boldsymbol{x}_0$$

- **Problem 2 (15 points):** Suppose that we perform an experiment to calculate the gravitational constant g as follows. We drop a ball from a certain height and measure its distance from the original point at certain time instants. The results of the experiment are shown in the following table:

Time (seconds)	1.00	2.00	3.00
Distance (meters)	5.00	19.5	44.0

  The equation relating the distance s and the time t at which s is measured is given by:

$$s = \frac{1}{2}gt^2$$

  1. Find a least squares estimate of g using the experimental results from the table above.(5 points)
  2. Suppose that we take an additional measurement at time $t = 4.00$ and obtain a distance of 78.5. Use the recursive least-squares algorithm to calculate the updated least squares estimate of g. Verify your solution by solving the problem from scratch with the new measurement and checking if you get the same answer. (10 points)

- **Problem 3 (15 points):** Suppose that a given speech signal $\{u_k \in \mathbb{R} : k = 1, \ldots, n\}$ is transmitted over a telephone cable with input-output behavior given by,

$$y_k = ay_{k-1} + bu_k + v_k,$$

  where, at each time k, $y_k \in \mathbb{R}$ is the output, $u_k \in R$ is the input (speech signal value) and v_k represents the white noise[1]. The parameters a, b are fixed known constants, and the initial condition is $y_0 = 0$.

[1] If $\boldsymbol{A}\boldsymbol{x} + \boldsymbol{w} = \boldsymbol{b}$, where $\boldsymbol{w}$ is a white noise vector, then the least squares estimate of $\boldsymbol{x}$ given $\boldsymbol{b}$ is the soltuion to the problem

$$\text{minimize } \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|^2.$$

Note than if $\boldsymbol{w}$ is a white noise vector, $\boldsymbol{D}\boldsymbol{w}$ (where $\boldsymbol{D}$ is a matrix) is not neccesarily a white noise vector.

We can measure the signal y_k at the output of the telephone cable, but we cannot directly measure the desired signal u_k or the noise signal v_k. Derive a formula for the linear least squares estimate of the signal $\{u_k, k = 1, \ldots, n\}$ given the signal $\{y_k, k = 1, \ldots, n\}$.

Global Search Algorithms

- **Problem 4 (40 points):** Consider the minimization of the following function:

$$f(x, y) = 3(1 - x)^2 e^{-x^2-(y+1)^2} - 10(x/5 - x^3 - y^5)e^{-x^2-y^2} - e^{-(x+1)^2-y^2}/3.$$

Implement the following algorithms in a programming language of your choice and and validate the algorithms by solving the problem above:

1. Naive Random Search (10 points)
2. Simulated Annealing (10 points)
3. Particle Swarm Optimization (10 points)
4. Genetic Algorithm (10 points)

for each algorithm, experiment with different hyperparameters (such as the choice of the neighbourhood in (1) and (2), number of particles in (3) or the choice of schema in (4)) and report the best working parameters.

Linear Programming

- **Problem 5 (20 points):** Consider a computer network consisting of six computers, A through F. The computers are connected according to the following links, with maximum data rates (in Mbps) shown: $AC(10)$, $BC(7)$, $BF(3)$, $CD(8)$, $DE(12)$, $DF(4)$. For example, $AC(10)$ means that the computers A and C are connected with a link that supports data rates up to 10 Mbps.

Suppose that A and B need to send data to E and F, respectively (no other communication takes place in then network). Any path through the given links above may be used as long as the path has no loop. Also, multiple paths (say from A to E) can be used simultaneously. Link bandwidth can be shared as long as the total data rate through the link does not exceed its maximum (the total data rate through a link is the sum of the data rates of communication in both directions).

For every Mbps of data rate the network can support for transmission from A to E, we receive 2 dollars. For every Mbps of data rate the network can support for transmission from B to F, we receive 3 dollars. Formulate a linear programming problem to represent the goal of maximizing the total revenue (10 points). Then, convert this problem into standard form (10 points).

Bonus Problem

- **Problem 6 (10 points):** Read about the Simplex algorithm using Chong's book or any other reference you can find. Implement the standard Simplex algorithm in a programming language of your choice. Verify that your algorithm works by selecting a linear programming problem (such as Problem 5 in this Pset), and comparing the solutions of your implementation with a solution obtained from a software package (such as linprog in MATLAB).