



STOR 512: OPTIMIZATION FOR ML AND NN
DEPARTMENT OF STATISTICS AND OPERATIONS RESEARCH

SPRING 2026

INSTRUCTOR: QUOC TRAN-DINH

HOMEWORK 1: INTRODUCTION TO OPTIMIZATION

Note: Please do not distribute this homework without instructor's permission.

Question 1. (25 points): Solve the following problems. In each case, find all optimal solutions.

- Minimize $f(x, y)$ over all positive real numbers x and y , with $f(x, y) = \frac{12}{x} + \frac{18}{y} + xy$.
- Maximize $f(x, y)$ over all positive real numbers x and y , with $f(x, y) = xy(72 - 3x - 4y)$.
- Find the largest and smallest value of $f(x, y, z) = 2x + 3y + 6z$ on all real numbers x, y , and z such that $x^2 + y^2 + z^2 = 1$.

Question 2. (25 points): Assume that you want to approximate a function $f(x, y, z) = x^2 + y^2 + z^2$ by a linear function $\ell(x, y, z) = a_1x + a_2y + a_3z + c$ such that $f(x, y, z) \approx \ell(x, y, z)$ at four points $(1, 0, 0)$, $(1, 1, 0)$, $(0, 1, 1)$, and $(1, 0, 1)$. Here, \approx means that $f(x, y, z)$ is almost equal to $\ell(x, y, z)$. Your goal is to find the coefficients a_1, a_2, a_3 , and c . Model this problem into an optimization problem of the form that minimizes the sum of squared errors between f and ℓ at these four points. Solve the resulting optimization problem.

Question 3. (25 points): There are 10 items with different weights and values given in the following table:

Weight [lb]	10	5	8	7	9	12	4	3	6	9
Value [\$]	9	8	12	6	5	11	7	2	4	8

Your goal is to select a few items from these 10 items such that the total weight does not exceed 40 lb, while maximizing the total value. Each item is selected at most one time.

Model this problem into an optimization problem and create a Python code to solve it. Run your Python code and report the results in detail. When submitting your solution, please also attach your code and a screenshot showing the results you ran.

Hint: This is a KnapSack Problem. You can use Dynamic Programming or Recursive method to solve it. Notice that you only need to show the optimal value, no need to show the optimal solution. This link might be useful: <https://www.geeksforgeeks.org/python-program-for-dynamic-programming-set-10-0-1-knapsack-problem/>

Question 4. (25 points): Consider a linear system of equations $Ax = b$, where

$$A = \begin{bmatrix} 8 & 4 & 1 & 2 \\ 4 & 10 & 1 & 3 \\ 1 & 1 & 4 & 1 \\ 2 & 3 & 1 & 8 \end{bmatrix} \quad \text{and} \quad b = \begin{bmatrix} 15 \\ 18 \\ 7 \\ 14 \end{bmatrix}.$$

Let

$$D = \begin{bmatrix} 8 & 0 & 0 & 0 \\ 0 & 10 & 0 & 0 \\ 0 & 0 & 4 & 0 \\ 0 & 0 & 0 & 8 \end{bmatrix},$$

be the diagonal of A and D^{-1} be the inverse of D .

The following iterative scheme is used to solve $Ax = b$:

$$x(t+1) = D^{-1}(D - A)x(t) + D^{-1}b, \quad t \geq 0,$$

where $x(0) = (0, 0, 0, 0)^\top$ is a given starting point, and $x(t)$ is the t -th iteration.

Code this method in Python and run it until the error between the t -th iterate $x(t)$ and the true solution x^* satisfies $\|x(t) - x^*\| \leq 10^{-6}$, where x^* is the solution of $Ax = b$. If it does not converge, then run it with at least 10^6 iterations. In the case it converges, then how many iterations does your method need? Plot the error $\|x(t) - x^*\|$ (in the y -axis) w.r.t. the iteration t (in the x -axis).