MAT 3007 Optimization Homework 1
Due: 11:59 pm on June 15, 2025
Solution

1. Consider the following optimization problem

$$\min f(x) \quad \text{s.t.} \quad x \in X,$$

where $f : \mathbb{R}^n \to \mathbb{R}$ and $X \subseteq \mathbb{R}^n$. For each of the following cases, give an example demonstrating that problem P may have an optimal solution, and an example demonstrating that P may not have an optimal solution, or argue that such an example does not exist.

(a) The function f is discontinuous and the set X is compact (compact means closed and bounded).
The problem P has an optimal solution if

$$X = [0, 1] \subseteq \mathbb{R}, \quad f(x) = \begin{cases} 1, & 0 < x \leq 1, \\ 0, & x = 0. \end{cases}$$

The problem P does not have an optimal solution if

$$X = [0, 1] \subseteq \mathbb{R}, \quad f(x) = \begin{cases} 1, & x = 0, \\ x, & 0 < x \leq 1. \end{cases}$$

In both cases, X is compact and f is discontinuous.

(b) The function f is continuous and the set X is not closed.
The problem P has an optimal solution if

$$X = [0, 1) \subseteq \mathbb{R}, \quad f(x) = x.$$

The problem P does not have an optimal solution if

$$X = (0, 1) \subseteq \mathbb{R}, \quad f(x) = x.$$

In both cases, X is not closed and f is continuous.

(c) The function f is linear and the set X is not bounded.
The problem P has an optimal solution if

$$X = \mathbb{R}, \quad f(x) = 0.$$

The problem P does not have an optimal solution if

$$X = \mathbb{R}, \quad f(x) = x.$$

In both cases, X is not bounded and f is linear.

(d) The function f is nonlinear and the set X is compact.
The problem P has an optimal solution if

$$X = [0,1], \quad f(x) = x^2.$$

The problem P does not have an optimal solution if

$$X = [0,1], \quad f(x) = \begin{cases} x, & 0 < x \leq 1, \\ 1, & x = 0. \end{cases}$$

In both cases, X is compact and f is nonlinear.

(e) The function f is linear and the set X is not closed.
The problem P has an optimal solution if

$$X = [0,1), \quad f(x) = x.$$

The problem P does not have an optimal solution if

$$X = (0,1), \quad f(x) = x.$$

In both cases, X is not closed and f is linear.

(f) The function f is linear and the set X is compact.
The problem P has an optimal solution if

$$X = [0,1], \quad f(x) = x.$$

In this case, if the set X is nonempty, then the problem P must have an optimal solution. Since the function f is linear, it is also continuous (you can verify this with the definition of continuity), and then by the Weierstrass Theorem, we know that a continuous function attains its maximum and minimum over a compact set.

2. For each of the following statements, state whether it is true or false. If true, provide a proof, and if false provide a counter-example.

(a) Any optimization problem whose feasible region is bounded must have an optimal solution.
False. Consider Problem 1 (a).

(b) Any optimization problem whose feasible region is unbounded cannot have an optimal solution.
False. Consider $\{\min \ x : x \geq 0\}$.

(c) Every global optimal solution to an optimization problem must have the same objective function value.
Ture. By definition; if not, one of the solutions cannot be globally optimal.

(d) I have solved an optimization problem and got an optimal solution x^*. Suppose now a constraint is added to the problem. If I find that x^* satisfies all remaining constraints, then x^* is an optimal solution of the modified problem.
True. Suppose it is not true, i.e. the new problem has a solution x' with a better objective value than that of x^*. Note that x' is feasible to the old problem, and since both problems have the same objective function, x' has a better objective value than x^* in the old problem. This contradicts the fact that x^* is an optimal solution of the old problem.

(e) Consider the following optimization problem:

$$\min[f(x)]^2 \quad \text{s.t.} \quad x \in X,$$

where $f(x)$ is a general function and X is a non-empty set. Suppose at a feasible solution $x^* \in X$ the objective value is 0, then x^* must be an optimal solution.
True. 0 is a lower bound of the problem.

(f) Consider the optimization problem

$$(P) : \min\{c^\top x : Ax = b\}$$

where $x \in \mathbb{R}^n$, $c \in \mathbb{R}^n$, $A \in \mathbb{R}^{m \times n}$, and $b \in \mathbb{R}^m$. If (P) has at least two feasible solutions with distinct objective function values then (P) is unbounded.
Suppose (P) has two feasible solutions x^1 and x^2 such that $c^\top x^1 < c^\top x^2$. Let $d = x^1 - x^2$. Then $c^\top d < 0$. Consider any point $x(\lambda) = x^2 + \lambda d$ for some $\lambda \geq 0$. Clearly

$$Ax(\lambda) = Ax^2 + \lambda Ad = Ax^2 + \lambda A(x^1 - x^2) = b.$$

Thus $x(\lambda)$ is a feasible solution. Now

$$c^\top x(\lambda) = c^\top x^2 + \lambda c^\top d.$$

Since $c^\top d < 0$, taking $\lambda \to +\infty$ we have $x(\lambda)$ remains feasible and $c^\top x(\lambda) \to -\infty$. Thus the problem is unbounded.

(g) Consider the optimization problem:

$$\min f(x) \quad \text{s.t.} \quad g(x) \leq 0.$$

Suppose the current optimal objective value is v. Now, if I change the right-hand side of the constraint to 1 and resolve the problem, the new optimal objective value will be less than or equal to v.
True. Changing the right-hand-side of the constraint to 1 enlarges the feasible region, and therefore may allow a smaller objective value than v.

3. Quantas Airways Ltd. must schedule its hundreds of reservation salesclerks around the clock to have at least r_t on duty during each 1-hour period starting at (24-hour) clock hour $t = 0, \ldots, 23$. A shift beginning at time t extends for 9 hours with 1 hour out for lunch in the fourth, fifth, or sixth hours of the shift. Shifts beginning at hour t cost the company c_t per day, including wages and night-hour premiums. Formulate an LP model to compute a minimum total cost daily shift schedule.

   Let x_t be the number of clerks working a shift starting at hour t and $y_{t,i}$ be the number of clerks working a shift starting at hour t who take lunch during hour i. The problem can be formulated as

$$\begin{aligned} \min \quad & \sum_{t=0}^{23} c_t x_t \\ \text{s.t.} \quad & \sum_{j=t-8}^{t} x_j - \sum_{j=t-5}^{t-3} y_{j,t} \geq r_t, \quad t = 0, \ldots, 23 \quad \text{(cover hour } t\text{)}, \\ & \sum_{i=t+3}^{t+5} y_{t,i} = x_t, \quad t = 0, \ldots, 23 \quad \text{(shift } t \text{ lunches)}, \\ & x_t, y_{t,i} \geq 0, \quad \forall t, \forall i, \end{aligned}$$

   where hour subscripts $j, t < 0$ are interpreted as $j+24, t+24$ and those $j, t > 23$ are interpreted as $j - 24, t - 24$.

4. Formulate the following optimization problems as linear programs or explain why you think it cannot be done.

   (a)

$$\begin{aligned} \min \quad & 2x_2 + |x_1 - x_3| \\ \text{s.t.} \quad & |x_1 + 2| + |x_2| \leq 5, \\ & x_3^2 \leq 1 \end{aligned}$$

   Let $y_1 = |x_1 - x_3|$, $y_2 = |x_1 + 2|$, $y_3 = |x_2|$.

$$
\begin{array}{rl}
\min & 2x_2 + y_1 \\
\text{s.t.} & y_1 \geq x_1 - x_3 \\
& y_1 \geq -x_1 + x_3 \\
& y_2 + y_3 \leq 5 \\
& y_2 \geq x_1 + 2 \\
& y_2 \geq -x_1 - 2 \\
& y_3 \geq x_2 \\
& y_3 \geq -x_2 \\
& x_3 \leq 1 \\
& x_3 \geq -1
\end{array}
$$

(b)

$$\min_{x_1,x_2} \{\max(|2x_1 + 3x_2|, |x_1 - x_2|) : |x_1| + 2\max(x_1, x_2) \leq 1\}$$

$$
\begin{array}{rl}
\min_{x,y} & y \\
\text{s.t.} & y \geq -(2x_1 + 3x_2) \\
& y \geq (2x_1 + 3x_2) \\
& y \geq (x_1 - x_2) \\
& y \geq -(x_1 - x_2) \\
& 3x_1 \leq 1 \\
& x_1 + 2x_2 \leq 1 \\
& -x_1 + 2x_2 \leq 1
\end{array}
$$

(c)

$$\min_{a,b} \left\{ \max_i (|y_i - ax_i - b|) : a \geq 0, b \geq 0 \right\}$$

$$
\begin{array}{rl}
\min_{t,a,b} & t \\
\text{s.t.} & t \geq (y_i - ax_i - b)\ \forall i \\
& t \geq -(y_i - ax_i - b)\ \forall i \\
& a \geq 0, b \geq 0
\end{array}
$$

5. Given a set of training data $\{x_i, y_i\}_{i=1,\ldots,N}$, where x_i is an n-dimensional feature vector and y_i is a label of value either 0 or 1. Think about each x_i representing a vector of lab test data of a patient i and y_i labels if this person has a certain disease. We want to build a linear classifier, i.e., a linear function

$$f(x) = \beta_0 + \sum_{j=1}^{n} \beta_j x_j,$$

so that for a given feature vector x, if $f(x) \geq 0.5$, then x is classified as $y = 1$, otherwise classified as $y = 0$.

A very popular method to build the classifier is called the absolute deviation regression (ADR). ADR is also called robust regression. The optimization model of ADR is described below.

$$\text{(ADR)} \qquad \min_{\beta_0,\ldots,\beta_n} \sum_{i=1}^{N} \left| y_i - \beta_0 - \sum_{j=1}^{n} \beta_j x_{ij} \right|,$$

where x_{ij} is the jth component of vector x_i. Notice that the ADR model is a nonlinear optimization problem.

Answer the following questions.

(a) Write a linear programming reformulation of (ADR).
The problem (ADR) can be formulated as a linear program with auxiliary variables t_i, $i = 1, \ldots, N$:

$$\begin{aligned} \min_{t_i, i=1,\ldots,N, \beta_0,\ldots,\beta_n} \quad & \sum_{i=1}^{N} t_i, \\ \text{s.t.} \quad & t_i \geq y_i - \beta_0 - \sum_{j=1}^{n} \beta_j x_{ij}, \quad \forall i = 1, \ldots, N, \\ & t_i \geq -(y_i - \beta_0 - \sum_{j=1}^{n} \beta_j x_{ij}), \quad \forall i = 1, \ldots, N. \end{aligned}$$

(b) Code your LP reformulation of (ADR) in CV, using the data file “regression.dat” provided.
We can write the CVXPY code as follows:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D
from cvxpy import *

# modified the regression.dat file to use read_table command
data = pd.read_table('regression.dat', delim_whitespace = True, header =

nData = 100
nFeature = 2

x = data.values[0: nData, :]
```

```
y = data.values[nData: 2 * nData, 0]

# b for beta
b = Variable(nFeature + 1)

# t for auxiliary variables for absolute values
t = Variable(nData)

constr = []
for i in range(nData):
constr += [t[i] >= y[i] - (b[0] + b[1] * x[i, 0] + b[2] * x[i, 1]),
-t[i] <= y[i] - (b[0] + b[1] * x[i, 0] + b[2] * x[i, 1])]

obj = sum(t)

prob = Problem(Minimize(obj), constr)
result = prob.solve()
```

An optimal solution is $(\beta_0, \beta_1, \beta_2)$ = (0.4036; 0.1850, -0.1995) (which may not be unique).

(c) Write a code to plot the data points and the hyperplane obtained from (ADR).
We can use the following python code:

```
b_sol = b.value
y_fit = np.ones(nData) * \
b_sol[0, 0] + x[:, 0] * b_sol[1, 0] + x[:, 1] * b_sol[2, 0]

x1_max = np.max(x[:, 0])
x1_min = np.min(x[:, 0])
x2_max = np.max(x[:, 1])
x2_min = np.min(x[:, 1])

mesh_x1, mesh_x2 = np.meshgrid(np.arange(x1_min, x1_max, 0.1),
np.arange(x2_min, x2_max, 0.1))

# calculate corresponding z values
plane_y = b_sol[0, 0] + mesh_x1 * b_sol[1, 0] + mesh_x2 * b_sol[2, 0]

fig = plt.figure()
ax = fig.add_subplot(111, projection='3d')
ax.scatter(x[:, 0], x[:, 1], y)
```

```
ax.plot_surface(mesh_x1, mesh_x2, plane_y)
```

The resulting figure shall look like the following one.