Optimization Homework 1

Hongyi Zhang

1、 I have already completed the equivalent courses at Southern University of Science and Technology (SUSTech), including the following courses.

Introduction to Computer Science (WashU) $\Longleftrightarrow$ Introduction to Computer Science (SUSTech)

Matrix Algebra (WashU) $\Longleftrightarrow$ Advanced Linear Algebra I and II (SUSTech)

Engineering Mathematics A $\Longleftrightarrow$ Mathematical Analysis I, II, III (SUSTech)

2. There are 5000 units of wood and 1500 units of labor.

nightstand : number — x_1 , price — \$100

chairs : number — x_2 , price — \$150

bookshelves : number — x_3 , price — \$200

dining tables : number — x_4 , price — \$400

$\Longrightarrow$ P (Profit of all products) $= 100x_1 + 150x_2 + 200x_3 + 400x_4$

and, $x_1 \geq 0$, $x_2 \geq 0$, $x_3 \geq 0$, $x_4 \geq 0$ (non-negative)

So, the objective function is to maximize P

$$\text{Maximize } P = \max(100x_1 + 150x_2 + 200x_3 + 400x_4)$$

Constraint Condition:

A nightstand $\longrightarrow$ 10 units of wood, 2 units of labor

A chair $\longrightarrow$ 12 units of wood, 4 units of labor

A bookshelf $\longrightarrow$ 25 units of wood, 8 units of labor

A dining table $\longrightarrow$ 20 units of wood, 12 units of labor

$\Longrightarrow$ Wood constraint: $10x_1 + 12x_2 + 25x_3 + 20x_4 \le 5000$

Labor constraint: $2x_1 + 4x_2 + 8x_3 + 12x_4 \le 1500$

$$\Longrightarrow \quad \text{Max } P = \text{Max}(100x_1 + 150x_2 + 200x_3 + 400x_4)$$

$$\begin{cases} 10x_1 + 12x_2 + 25x_3 + 20x_4 \le 5000 \\ 2x_1 + 4x_2 + 8x_3 + 12x_4 \le 1500 \\ x_1, x_2, x_3, x_4 \ge 0 \end{cases}$$

This problem falls into the class of Linear Programming problems. And, it is linear optimization problem because both the objective function and the constraints are linear functions.

Finally, I applied MATLAB to find the solution as follows

编辑器 - C:\Users\Zhy20\Desktop\MATLAB\Optimization\Homework01.m

Homework01.m

```
% Define the coefficients of the objective function (need to be inverted to convert into a minimization problem)
f = -[100 150 200 400];

% Define the coefficient matrix and the right-hand side values of the inequality constraints
A = [10 12 25 20; 2 4 8 12];
b = [5000; 1500];

% Define the limits of the variables (here the lower limit of all variables is 0, and the upper limit defaults to infinity)
lb = [0 0 0 0];
ub = [];

% Set variable types for integer programming, assuming all variables are integers
intcon = 1:4;

% Call intlinprog function to solve
[x, fval, exitflag, output] = intlinprog(f, intcon, A, b, [], [], lb, ub);

disp(x)
```

The solution is $x_1 = 372$, $x_2 = 3$, $x_3 = 0$, $x_4 = 62$

Therefore, the max sale profit is

$$372 \times 100 + 3 \times 150 + 0 \times 200 + 62 \times 400 = \$62450$$

3. Let the center of the circle be (a,b) and the radius be r.

The goal is to find the smallest radius r such that all point (x_i, y_i) are within or on the circle.

$\Longrightarrow$ ① $\min r$ subject to $(x_i-a)^2+(y_i-b)^2 \le r^2$ for all $i=1,\cdots,N$

$\Longrightarrow$ ② $\min r^2$ subject to $(x_i-a)^2+(y_i-b)^2 \le r^2$ for all $i=1,\cdots,N$

Also, we can have a $2\times N$ matrix $A = \begin{pmatrix} x_1 & x_2 & \cdots & x_N \\ y_1 & y_2 & \cdots & y_N \end{pmatrix}$

According a paper call The Minimum Covering Sphere Problem, by D.J, Elzinga and D.W. Hearn, let $m=N$, $n=2$, this smallest circle problem can be formulated as a quadratic programming problem.

Based on Kuhn-Tucker conditions, which are both necessary and sufficient for ②, assure the existence of multipliers V_i, $i=1\cdots N$ such that $\sum_{i=1}^{N} V_i = 1$, $V_i \ge 0$, $i=1,\cdots N$ and $\vec{V}' = (V_1, V_2, \cdots, V_N)$

$$\Longrightarrow \min\left[\vec{V}^T(A^TA)\vec{V} - \sum_{i=1}^{N} V_i(x_i^2+y_i^2)\right] = \min f(\vec{V})$$

$$\text{subject to } \sum_{i=1}^{N} V_i = 1, \ V_i \ge 0, \text{ for } i=1,\cdots,N$$

If $\bar{V} = (\bar{V}_1, \bar{V}_2, \cdots, \bar{V}_N)$ is the solution of this quadratic programming problem, then the center of the circle is located at (a,b) where

$$\begin{cases} a = \bar{V}_1 x_1 + \bar{V}_2 x_2 + \cdots \bar{V}_N x_N \\ b = \bar{V}_1 y_1 + \bar{V}_2 y_2 + \cdots \bar{V}_N y_N \end{cases}$$

and the radius of the circle is given by $r = \sqrt{-f(\bar{V})}$

And, there is another solution method call Welzl's Algorithm.

The following part is the pseudocode of this Algorithm.

```
def welzl(points):
  """
  compute the minimum bounding sphere for a given set of points.
  points: point set
  return value: center and radius of the smallest enclosing circle
  """

  # If the point set is empty, an empty bounding sphere is returned.
  if not points:
    return None

  # If the point set has only one point, then that point is the center of the smallest enclosing sphere, with a radius of 0.
  if len(points) == 1:
    return Ball(points[0], 0)

  # A point is randomly chosen as the center of the minimum bounding sphere.
  center = random.choice(points)

  # Divide the remaining points into two groups: points inside the minimum bounding sphere and points outside the minimum bounding sphere.
  inside = []
  outside = []
  for point in points:
    if distance(point, center) <= radius:
      inside.append(point)
    else:
      outside.append(point)

  # If all points are within the minimum bounding sphere, then the minimum bounding sphere is the final result.
  if not outside:
    return Ball(center, radius)

  # Calculate the minimum enclosing sphere recursively among the remaining points.
  ball = welzl(outside)

  # If the recursively obtained minimum bounding sphere contains the center point, then the minimum bounding sphere is the final result.
  if center in ball:
    return ball

  # Otherwise, add the center point to the minimum bounding sphere and return the expanded minimum bounding sphere.
  else:
    return Ball(center, radius + distance(center, ball.center))
```

4. (a) Let $n=1$.

$$f(x) = f(x^*) + f'(x^*)(x-x^*) + \frac{1}{2}f''(z)(x-x^*)^2$$

where z is a point between x and x^*

Since $f'(x^*)=0$

$$f(x) = f(x^*) + \frac{1}{2}f''(z)(x-x^*)^2$$

$$\Rightarrow f(x) - f(x^*) = \frac{1}{2}f''(z)(x-x^*)^2$$

Since $f''(x) > 0$ for all $x \in I$,

Therefore, $f(x) - f(x^*) > 0$ for all $x \neq x^*$

This inequality implies that $f(x) > f(x^*)$ for all $x \neq x^*$, which means that x^* is a strict global minimizer of f on the open interval I.

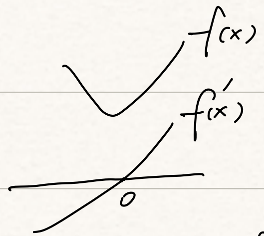
(b) Let $f(x) = e^{x^2}$,

We can have $f'(x) = 2x e^{x^2}$

$$f''(x) = 2e^{x^2} + 4x^2 e^{x^2} = 2e^{x^2}(1+2x^2)$$

Setting $f'(x^*) = 0$

$$2x^* e^{x^{*2}} = 0 \implies x^* = 0$$



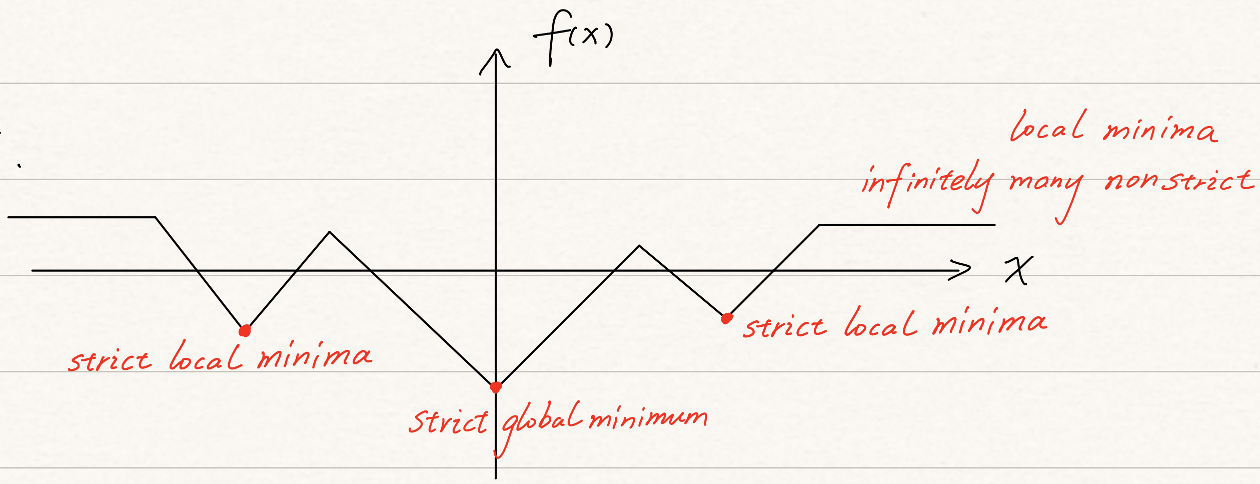
$$f''(x) = 2e^{x^2}(1+2x^2) > 0 \text{ for all } x \in R$$

$f'(0) = 0$, so $x^* = 0$ is a stationary point

$f''(x) > 0$ and $f'(0) = 0$ implies that $f(x)$ is strictly convex

Therefore, $x^* = 0$ is a strict global minimizer of $f(x) = e^{x^2}$ on R

5.



6. (a) $f: R^3 \to R^2$, $f(x,y,z) = (xz+y^2, \sin(xyz)+z)$

Suppose $f_1(x,y,z) = xz+y^2$,

$$\frac{\partial f_1}{\partial x} = z, \quad \frac{\partial f_1}{\partial y} = 2y, \quad \frac{\partial f_1}{\partial z} = x$$

Suppose $f_2(x,y,z) = \sin(xyz)+z$

$$\frac{\partial f_2}{\partial x} = yz\cos(xyz), \quad \frac{\partial f_2}{\partial y} = xz\cos(xyz), \quad \frac{\partial f_2}{\partial z} = xy\cos(xyz)+1$$

$$J_f(x,y,z) = \begin{pmatrix} \frac{\partial f_1}{\partial x} & \frac{\partial f_1}{\partial y} & \frac{\partial f_1}{\partial z} \\ \frac{\partial f_2}{\partial x} & \frac{\partial f_2}{\partial y} & \frac{\partial f_3}{\partial z} \end{pmatrix} = \begin{pmatrix} z & 2y & x \\ yz\cos(xyz) & xz\cos(xyz) & xy\cos(xyz+1 \end{pmatrix}$$

$$J_f(0,-1,1) = \begin{pmatrix} 1 & -2 & 0 \\ -1 & 0 & 1 \end{pmatrix}$$

(b) Suppose that there is a direction $\vec{v} = (v_1, v_2, v_3)^T$ in which the directional derivative of f at $(0,-1,1)$ is zero.

$$D_{\vec{v}} f(0,-1,1) = J_f(0,-1,1) \cdot \vec{v}$$

$$= \begin{pmatrix} 1 & -2 & 0 \\ -1 & 0 & 1 \end{pmatrix} \begin{pmatrix} v_1 \\ v_2 \\ v_3 \end{pmatrix}$$

$$= \begin{pmatrix} v_1 - 2v_2 \\ -v_1 + v_3 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

$$\Rightarrow \begin{cases} v_1 - 2v_2 = 0 \\ -v_1 + v_3 = 0 \end{cases} \Rightarrow \begin{cases} v_1 = 2v_2 \\ v_1 = v_3 \end{cases}$$

Therefore, the direction vector $\vec{v}$ that satisfies this condition is any vector of the form $\vec{v} = (2v_2, v_2, 2v_2)$ where v_2 can be any real number, forexample $(2, 1, 2)$