

TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN, ĐHQG-HCM



# BÀI TỔNG KẾT MÔN QUY HOẠCH TUYỂN TÍNH

**Giảng viên: Nguyễn Lê Hoàng Anh**

Nhóm TVN: Phan Hồng Trâm - 21110414

Nguyễn Hoàng Yến Như - 21110359

Nguyễn Phạm Tiểu Vy - 21110457

## Mục lục

<b>1</b>	<b>Thông tin nhóm, nhiệm vụ, mức độ hoàn thành, ký tên</b>	<b>2</b>
1.1	Thông tin nhóm . . . . .	2
1.2	Nhiệm vụ và mức độ hoàn thành . . . . .	2
1.3	Ký tên . . . . .	2
<b>2</b>	<b>Thông tin bài làm</b>	<b>3</b>
2.1	Bài 1. Sử dụng thư viện <b>CVXOPT</b> bằng cách sử dụng hàm <code>solvers.lp()</code> . . . . .	3
2.2	Bài 2: Sử dụng phương pháp hình học (graphical method) để giải bài toán quy hoạch tuyến tính 2 biến. . . . .	4
<b>3</b>	<b>Hướng dẫn code</b>	<b>5</b>
3.1	Bài 1: Hướng dẫn nhập dữ liệu với thư viện <b>CVXOPT</b> . . . . .	5
3.2	Bài 2: Hướng dẫn nhập dữ liệu với phương pháp hình học . . . . .	8
<b>4</b>	<b>Tài liệu tham khảo</b>	<b>11</b>

## 1 Thông tin nhóm, nhiệm vụ, mức độ hoàn thành, ký tên

### 1.1 Thông tin nhóm

**Tên nhóm:** TVN

**Thành viên:**

1. Phan Hồng Trâm (Trưởng nhóm)
  - (a) MSSV: 21110414
  - (b) Chuyên ngành: Toán tin (Khoa học dữ liệu)
2. Nguyễn Hoàng Yến Như
  - (a) MSSV: 21110359
  - (b) Chuyên ngành: Toán Ứng Dụng (Sư phạm)
3. Nguyễn Phạm Tiểu Vy
  - (a) MSSV: 21110457
  - (b) Chuyên ngành: Toán Ứng Dụng (Sư phạm)

### 1.2 Nhiệm vụ và mức độ hoàn thành

**Phân công nhiệm vụ:**


1. Phan Hồng Trâm: code chương trình, sửa lỗi code.
2. Nguyễn Hoàng Yến Như: tìm nguồn tài liệu, giải thích công thức toán, test chương trình.
3. Nguyễn Phạm Tiểu Vy: ghi file báo cáo, test chương trình.

**Mức độ hoàn thành:** Các bạn đã hoàn thành nhiệm vụ mà trưởng nhóm giao.

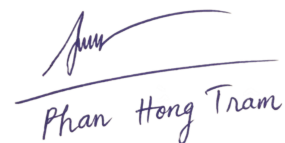
### 1.3 Ký tên



((a)) Nguyễn Phạm Tiểu Vy

  
Nguyễn Hoàng Yến Như

((b)) Nguyễn Hoàng Yến Như

  
Phan Hong Tram

((c)) Phan Hồng Trâm

Hình 1: Chữ ký các bạn trong nhóm

## 2 Thông tin bài làm

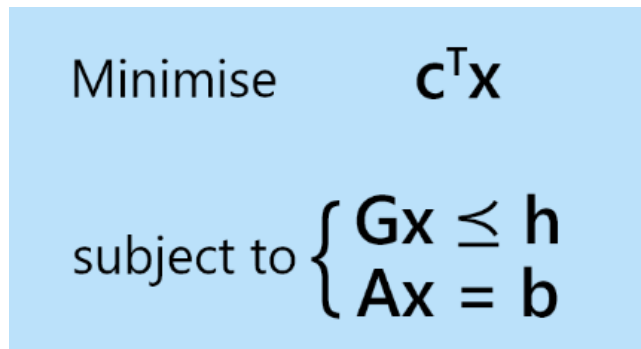
Bài làm gồm 2 phần và 2 file như sau:

### 2.1 Bài 1. Sử dụng thư viện **CVXOPT** bằng cách sử dụng hàm `solvers.lp()`.

- **Làm được:** Có thể giải bài toán quy hoạch tuyến tính với số biến tùy ý, nhưng phải nhập biến vào đúng định dạng của hàm `solvers.lp()`.

- **Chưa làm được:** Chưa thể giải bài toán ở dạng nhập tùy ý.

1. Chương trình chỉ có thể giải bài toán quy hoạch tuyến tính với dạng tổng quát như sau:


$$\begin{array}{ll} \text{Minimise} & \mathbf{c}^T \mathbf{x} \\ \text{subject to} & \begin{cases} \mathbf{G}\mathbf{x} \leq \mathbf{h} \\ \mathbf{A}\mathbf{x} = \mathbf{b} \end{cases} \end{array}$$

Hình 2: Dạng tổng quát để nhập vào chương trình

Trong đó:

- (a)  $\mathbf{x}$  là vector biến số, chính là nghiệm mà ta cần giải.
  - (b) Vector  $\mathbf{c}$ ,  $\mathbf{h}$ ,  $\mathbf{b}$  và ma trận  $\mathbf{G}$ ,  $\mathbf{A}$  là đầu vào của hàm `solvers.lp(c, G, h, A, b)`.
2. Ta phải nhập đầu vào là các ma trận và vector, nếu hàm mục tiêu ở dạng **max** ta phải đổi về **min**.
  3. Nếu ràng buộc bất đẳng thức có dấu  $\geq$  thì ta phải đổi về dấu  $\leq$ .

- **Bố cục của file code:**

1. Hàm `input_matrix`: Nhập vào ma trận theo từng cột, các tham số ngăn cách nhau bởi dấu cách (space), sau đó dùng hàm `matrix` của thư viện `CVXOPT` để trả về các ma trận và vector đúng dạng cần nhập vào của bài toán.
2. Hàm `solve`: Dùng thư viện `CVXOPT` để giải bài toán quy hoạch tuyến tính và trả ra nghiệm tối ưu, giá trị tối ưu và số bước lặp.

## 2.2 Bài 2: Sử dụng phương pháp hình học (graphical method) để giải bài toán quy hoạch tuyến tính 2 biến.

- **Làm được:** Có thể nhập dấu của bất đẳng thức tùy ý.

- **Chưa làm được:**

1. Phải nhập hàm mục tiêu ở dạng **min**.
2. Không giải được các bài toán không giới nội.

- **Bố cục của bài code:**

1. Hàm `get_objective_function()`: Nhập hệ số cho hàm mục tiêu.
2. Hàm `get_constraints()`: Nhập hệ số và dấu bất đẳng thức cho ràng buộc.
3. Giải hệ phương trình của từng đường thẳng trong ràng buộc bằng thư viện `sympy`.
4. Lọc ra cái điểm không thỏa mãn ràng buộc và vẽ miền chấp nhận được bằng thư viện `matplotlib` và hàm `ConvexHull` của thư viện `scipy.spatial`.
5. Dùng hàm `linprog` của thư viện `scipy.optimize` với `method = 'simplex'` (phương pháp đơn hình) để tìm nghiệm tối ưu và thư viện `matplotlib` để vẽ nghiệm tối ưu.
  - (a) Nếu có nghiệm, chương trình sẽ hỏi có muốn in ra nghiệm tối ưu, giá trị tối ưu, số bước lặp không và vẽ hình.
  - (b) Nếu không có nghiệm, chương trình sẽ in ra thông báo 'No feasible solution' và vẽ hình.

**Lưu ý:** vì hàm `linprog` được thiết kế để tìm nghiệm tối ưu (optimal solution) của các bài toán minimize, nên lúc nhập dữ liệu ta sẽ đổi hàm mục tiêu về dạng min. Và sau khi tìm ra nghiệm tối ưu thì sẽ có thêm 1 option hỏi rằng "What type of the objective function? (max/min): " để tính ra giá trị tối ưu (optimal value) của bài toán gốc.

### 3 Hướng dẫn code

Cả 2 file code đều được code bằng ngôn ngữ Python. Các thư viện được dùng trong cả 2 file code:

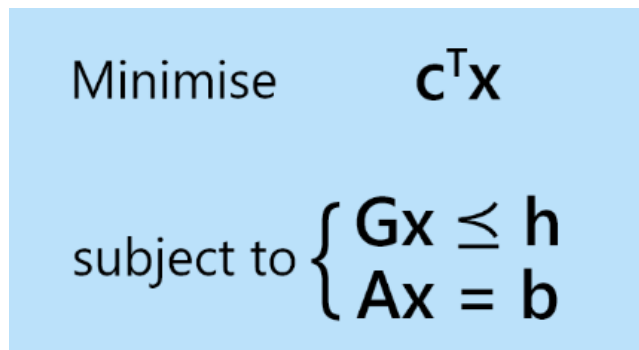
```
# Bai 1: Dung CVXOPT
from cvxopt import matrix, solvers
import numpy as np
# Bai 2: Phuong phap hinh hoc (graphical method)
from sympy import symbols, Eq, solve, N # tim nghiem de ve hinh
import numpy as np
import matplotlib.pyplot as plt # ve hinh
from scipy.optimize import linprog # tim nghiem toi uu bang phuong phap don hinh
from scipy.spatial import ConvexHull # Convex hulls in N dimensions.
import warnings
warnings.filterwarnings("ignore")
```

Nếu chưa import thư viện, ta có thể dùng cú pháp `pip install <library_name>` trong Command Prompt.

#### 3.1 Bài 1: Hướng dẫn nhập dữ liệu với thư viện **CVXOPT**

CVXOPT là một thư viện miễn phí trên Python giúp giải bài toán trong Convex Optimization - Linear Programming.

- Ta đưa bài toán cần giải về dạng tổng quát:



$$\begin{array}{ll} \text{Minimise} & \mathbf{c}^T \mathbf{x} \\ \text{subject to} & \begin{cases} \mathbf{G}\mathbf{x} \leq \mathbf{h} \\ \mathbf{A}\mathbf{x} = \mathbf{b} \end{cases} \end{array}$$

Hình 3: Dạng tổng quát của LP

Trong đó:  $\mathbf{G} \in \mathbb{R}^{m \times n}$ ,  $\mathbf{h} \in \mathbb{R}^m$  và  $\mathbf{A} \in \mathbb{R}^{p \times n}$ ,  $\mathbf{b} \in \mathbb{R}^p$  và  $\mathbf{c}, \mathbf{x} \in \mathbb{R}^n$

- Ví dụ:

$$\begin{cases} \max & 3x_1 + 2x_2 \\ & 2x_1 + x_2 \leq 8 \\ & -x_1 + x_2 \leq 1 \\ & x_2 \leq 2 \\ & x_1 \geq 0 \\ & x_2 \geq 0 \end{cases}$$

Các điều kiện ràng buộc có thể viết lại dưới dạng  $\mathbf{G}\mathbf{x} \preceq \mathbf{h}$ , trong đó:

$$\mathbf{G} = \begin{bmatrix} 2 & 1 \\ -1 & 1 \\ 0 & 1 \\ -1 & 0 \\ 0 & -1 \end{bmatrix}, \quad \mathbf{h} = \begin{bmatrix} 8 \\ 1 \\ 2 \\ 0 \\ 0 \end{bmatrix}$$

và  $\mathbf{c}^T = \begin{bmatrix} -3 \\ -2 \end{bmatrix}$  vì ta đưa bài toán dạng **max** về **min**.

- **Cách nhập vào file code:** Ta lần lượt nhập các ma trận/vector  $\mathbf{c}$ ,  $\mathbf{G}$  và  $\mathbf{h}$  theo từng cột, ngăn cách nhau bởi dấu cách (space):

– Nhập ma trận  $\mathbf{c}$ :

```
Input matrix c (Nhập ma trận/vector hệ số hàm mục tiêu):
Enter the number of rows of matrix/vector c (Nhập số lượng hệ số trên hàm mục tiêu):2
Enter the entries in a single column (separated by space) (Nhập ma trận/vector theo cột):
3 2
matrix c_T is:
[ 3.00e+00]
[ 2.00e+00]
```

Hình 4: Nhập ma trận  $\mathbf{c}$

– Nhập ma trận  $\mathbf{G}$ :

```
Input matrix G (Nhập ma trận hệ số BĐT ràng buộc):
Enter the number of rows of matrix G (Nhập số hàng của ma trận G (Số lượng constraints)):5
Enter the number of columns of matrix G (Nhập số cột của ma trận G (Số lượng biến)):2
Enter the entries in a single column (separated by space) (Nhập ma trận theo từng cột):
2 -1 0 -1 0 1 1 1 0 -1
matrix G is:
[ 2.00e+00 1.00e+00]
[-1.00e+00 1.00e+00]
[ 0.00e+00 1.00e+00]
[-1.00e+00 0.00e+00]
[ 0.00e+00 -1.00e+00]
```

Hình 5: Nhập ma trận  $\mathbf{G}$

– Nhập ma trận  $\mathbf{h}$ :

```
Input matrix h (Nhập ma trận/vector giá trị vế phải BĐT ràng buộc):
Enter the number of rows of matrix h (Nhập số hàng của ma trận h (Số lượng constraints)):5
Enter the entries in a single column (separated by space) (Nhập ma trận/vector theo cột):
8 1 2 0 0
matrix b is:
[ 8.00e+00]
[ 1.00e+00]
[ 2.00e+00]
[ 0.00e+00]
[ 0.00e+00]
```

Hình 6: Nhập ma trận  $\mathbf{h}$

- Sau đó chương trình sẽ hỏi ta có muốn in ra bước làm hay không?

```
Do you want to show process? [Y/N]
```

- Nếu có, chương trình in ra bước làm:

```
Do you want to show process? [Y/N] Y
      pcost      dcost      gap      pres      dres      k/t
0: -1.0826e+01 -2.6609e+01 2e+01 3e-01 8e-01 1e+00
1: -1.2608e+01 -1.4163e+01 1e+00 4e-02 9e-02 3e-01
2: -1.2965e+01 -1.3046e+01 6e-02 2e-03 4e-03 1e-02
3: -1.3000e+01 -1.3000e+01 6e-04 2e-05 4e-05 1e-04
4: -1.3000e+01 -1.3000e+01 6e-06 2e-07 4e-07 1e-06
5: -1.3000e+01 -1.3000e+01 6e-08 2e-09 4e-09 1e-08
Optimal solution found.
Optimal solution:
[ 3.00e+00  2.00e+00]

optimal value:
-12.999999965548074
Number of iterations:
5
```

- Ngược lại nếu không, chương trình chỉ in ra Optimal Solution, Optimal Value và Number of iterations:

```
Do you want to show process? [Y/N] N
Optimal solution:
[ 3.00e+00  2.00e+00]

Optimal value:
-12.999999965548074
Number of iterations:
5
```



### 3.2 Bài 2: Hướng dẫn nhập dữ liệu với phương pháp hình học

Ta nhập như bình thường, nhưng hàm mục tiêu thì nhớ phải đổi sang hàm min.

Ví dụ:

$$\begin{cases} \max 3x_1 + 2x_2 \\ 2x_1 + x_2 \leq 8 \\ -x_1 + x_2 \leq 1 \\ x_2 \leq 2 \\ x_1 \geq 0 \\ x_2 \geq 0 \end{cases}$$

- Đầu tiên ta nhập hệ số của hàm mục tiêu (Lưu ý nhớ đổi sang dạng min):

```
Enter the coefficient for x in the objective function: -3
Enter the coefficient for y in the objective function: -2
```

- Sau đó nhập số lượng ràng buộc (ở ví dụ trên ta có 5 ràng buộc):

```
Enter the number of constraints: 5
```

- Bắt đầu nhập các hệ số của x và y, dấu ( $\leq$  hoặc  $\geq$ ) và giá trị  $b_i$  trong các ràng buộc:

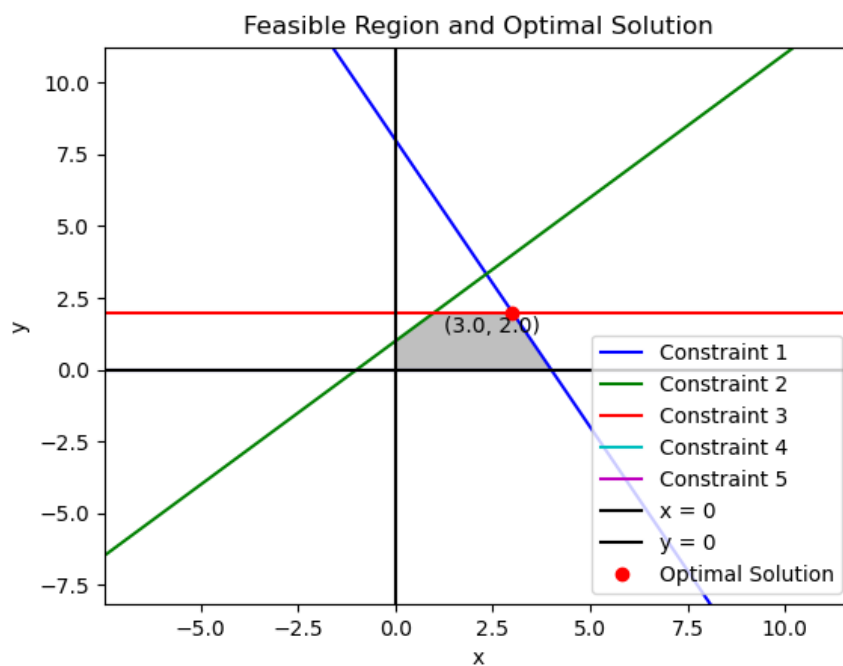
```
Enter the coefficient for x in the constraint: 2
Enter the coefficient for y in the constraint: 1
Enter the inequality sign (<= or >=): <=
Enter the right-hand side value: 8
Enter the coefficient for x in the constraint: -1
Enter the coefficient for y in the constraint: 1
Enter the inequality sign (<= or >=): <=
Enter the right-hand side value: 1
Enter the coefficient for x in the constraint: 0
Enter the coefficient for y in the constraint: 1
Enter the inequality sign (<= or >=): <=
Enter the right-hand side value: 2
Enter the coefficient for x in the constraint: 1
Enter the coefficient for y in the constraint: 0
Enter the inequality sign (<= or >=): >=
Enter the right-hand side value: 0
Enter the coefficient for x in the constraint: 0
Enter the coefficient for y in the constraint: 1
Enter the inequality sign (<= or >=): >=
Enter the right-hand side value: 0
```

- Sau đó chương trình hỏi rằng có muốn in ra số bước lặp, nghiệm và giá trị tối ưu không:

```
Do you want to print number of iterations, optimal point and optimal value? (yes/no):
```

- Nếu có, chương trình in ra theo yêu cầu đồng thời vẽ hình:

```
Do you want to print number of iterations, optimal point and optimal value? (yes/no): yes
What type of the objective function? (max/min): max
optimal_point: [3. 2.]
optimal_value: 13.0
Total number of iterations: 3
```



- Nếu không, chương trình chỉ vẽ hình.

Ví dụ với bài toán vô nghiệm:

$$\begin{cases} \min x_1 + x_2 \\ x_1 + 2x_2 \geq 2 \\ 3x_1 + 2x_2 \leq 1 \\ x_1 + x_2 \geq 1 \\ x_1 \geq 0 \\ x_2 \geq 0 \end{cases}$$

- Đầu tiên ta nhập hệ số của hàm mục tiêu (Lưu ý nhớ đổi sang dạng min):

```
Enter the coefficient for x in the objective function: 1
Enter the coefficient for y in the objective function: 1
```

- Sau đó nhập số lượng ràng buộc (ở ví dụ trên ta có 5 ràng buộc):

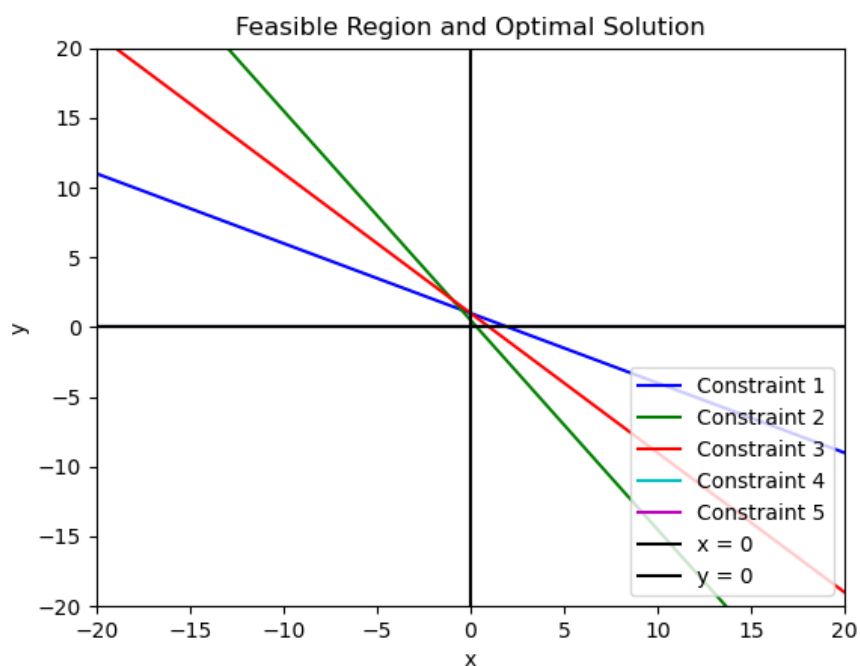
```
Enter the number of constraints: 5
```

- Bắt đầu nhập các hệ số của x và y, dấu ( $\leq$  hoặc  $\geq$ ) và giá trị  $b_i$  trong các ràng buộc:

```
Enter the coefficient for x in the constraint: 1
Enter the coefficient for y in the constraint: 2
Enter the inequality sign (<= or >=): >=
Enter the right-hand side value: 2
Enter the coefficient for x in the constraint: 3
Enter the coefficient for y in the constraint: 2
Enter the inequality sign (<= or >=): <=
Enter the right-hand side value: 1
Enter the coefficient for x in the constraint: 1
Enter the coefficient for y in the constraint: 1
Enter the inequality sign (<= or >=): >=
Enter the right-hand side value: 1
Enter the coefficient for x in the constraint: 1
Enter the coefficient for y in the constraint: 0
Enter the inequality sign (<= or >=): >=
Enter the right-hand side value: 0
Enter the coefficient for x in the constraint: 0
Enter the coefficient for y in the constraint: 1
Enter the inequality sign (<= or >=): >=
Enter the right-hand side value: 0
```

- Vì hàm vô nghiệm nên chương trình hiển thị không tìm thấy nghiệm và vẽ hình:

```
No feasible solution
```



## 4 Tài liệu tham khảo

### Tài liệu

- [1] [4 Ways to Solve Linear Programming in Python](#)
- [2] [GIẢI TOÁN TỐI ƯU VỚI PYTHON](#)
- [3] [Solving a linear program - CVXOPT](#)
- [4] [The solvers module in SymPy implements methods for solving equations](#)
- [5] [Linear programming solves by scipy](#)
- [6] [Convex Optimization Problems](#)