TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN

BÀI BÁO CÁO CODE QUY HOẠCH TUYẾN TÍNH

Thành viên:

1. Trần Quang Thuận 20280093
2. Huỳnh Bá Thiện 20280089

Mục lục

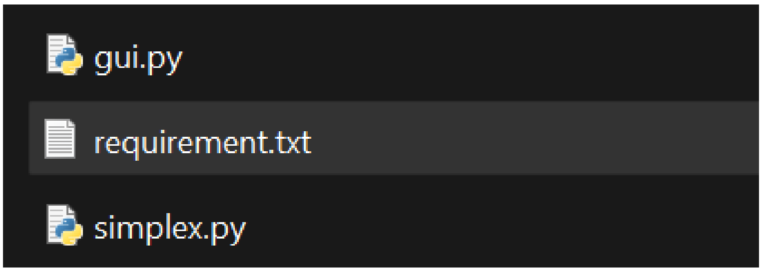
Ứng dụng giải bài toán quy hoạch tuyến tính

	MSSV	Họ và tên	Email	Nhóm trưởng	NV thực hiện	Đánh giá	Ký tên
1	20280093	Trần Quang Thuận	20280093@ student.hcmus .edu.vn	X	Thuật toán và Tổng hợp báo cáo	Khá tốt (9) chưa in hoàn chỉnh phần vô số nghiệm	
2	20280089	Huỳnh Bá Thiện	20280089@ student.hcmus .edu.vn		GUI và báo cáo	Tốt (10)	

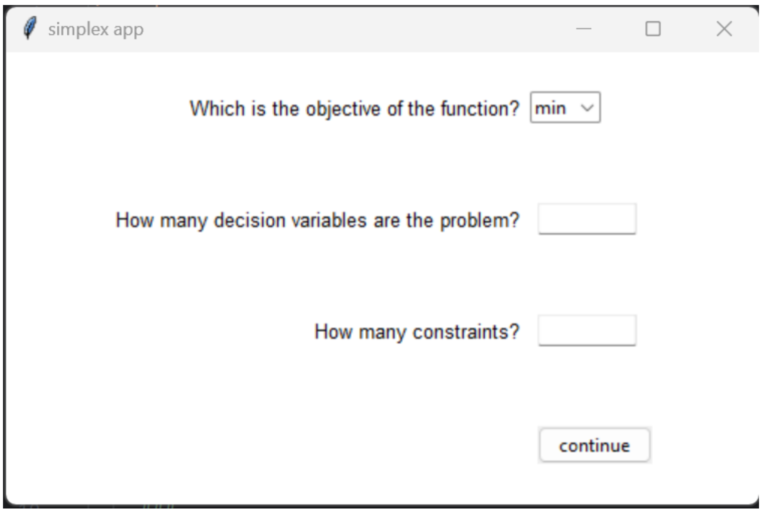
Tóm tắt nội dung

Thông qua quá trình học, để giải quyết một bài toán quy hoạch tuyến tính cần rất nhiều phương pháp. Bài báo cáo này chỉ tập trung vào 3 thuật toán chính đơn hình Dantzig, Bland và đơn hình 2 pha.

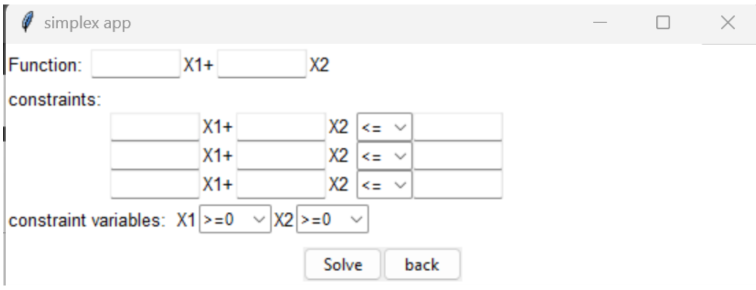
1 Hướng dẫn sử dụng code



Gồm hai file .py. một file chứa cái thuật toán quy hoạch tuyến tính, một file chứa GUI của thuật toán và file requirement.txt chứa các thư viện cần thiết để chạy chương trình
Khi sự dụng ta chỉ cần mở file gui.py. Sau đó dùng bất kỳ editor để chạy file gui.py
Lưu ý: cần cài những thư viện đã dược để trong file requirement.txt



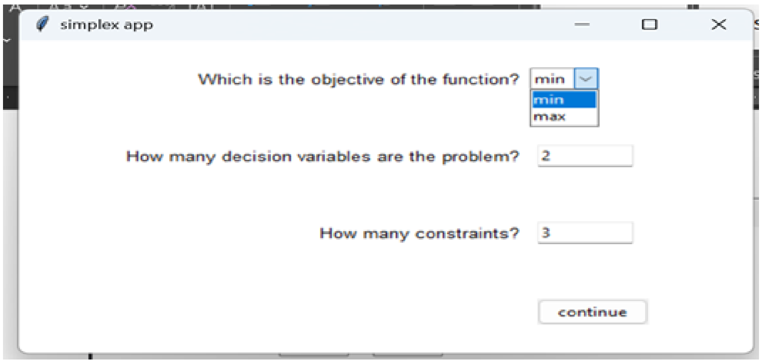
Giao diện sau khi chạy code. Ở đây ta cần số biến, số ràng buộc và max(hoặc min) theo để bài. Sau đó continue



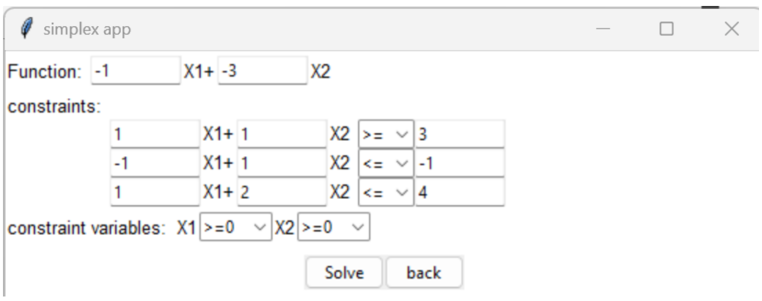
Sau khi nhấn continue ta sẽ có giao diện mới. Ở giao diện này ta sẽ nhập từng hệ số a, b, c và ràng buộc dấu cho biến. cuối cùng nhấn solve để giải.
Hoặc nếu trường hợp có nhập nhầm về số biến và ràng buộc thì cũng sao chúng ta có nút back để trở về ban đầu.

Ví dụ minh họa:

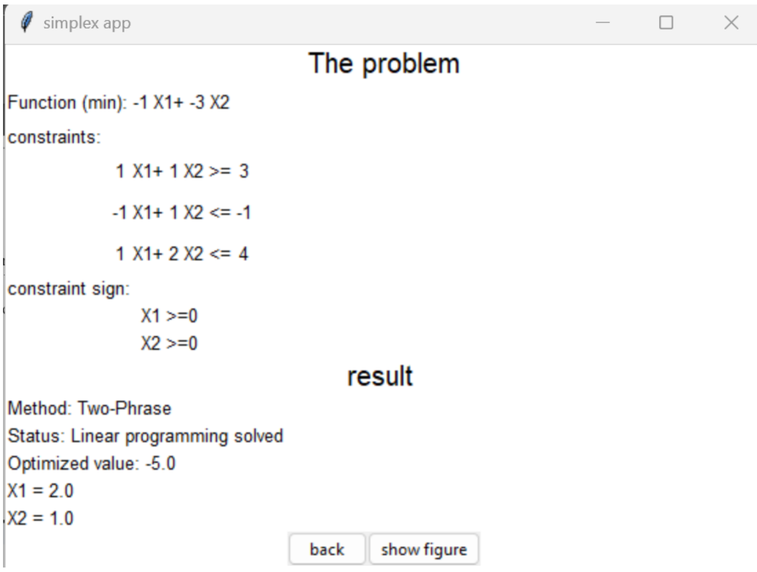
$$
\begin{cases}
min - x_1 - 3x_2 \\
x_1 + x_2 \geq 3 \\
-x_1 + x_2 \leq -1 \\
x_1 + x_2 \leq 4
\end{cases}
$$



Ta ta chọn số biến là 2, số ràng buộc là 3, với hàm mục tiêu là min. nhấn continue



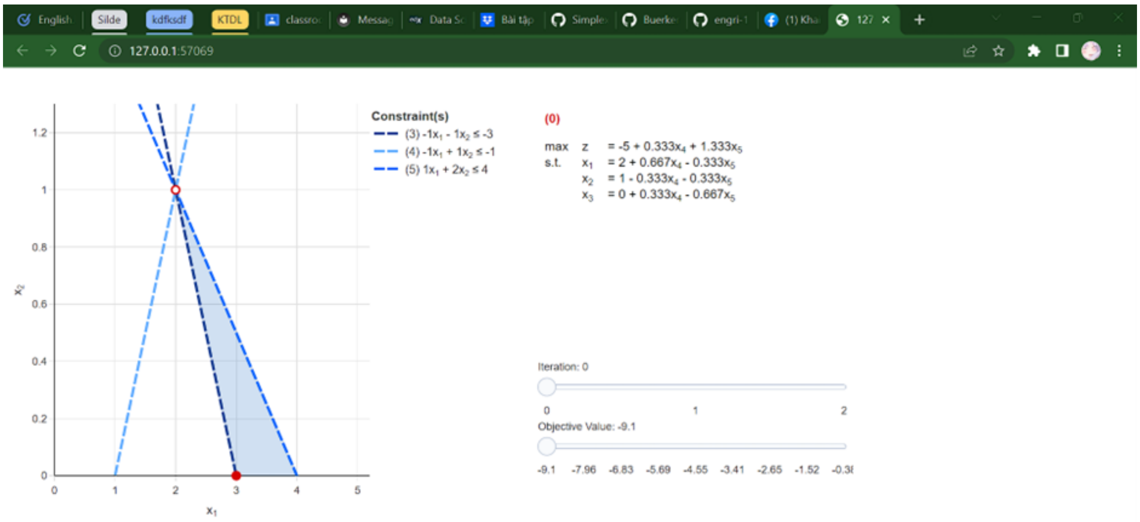
Ta điền các hệ số theo đề bài và nhấn solve để giải



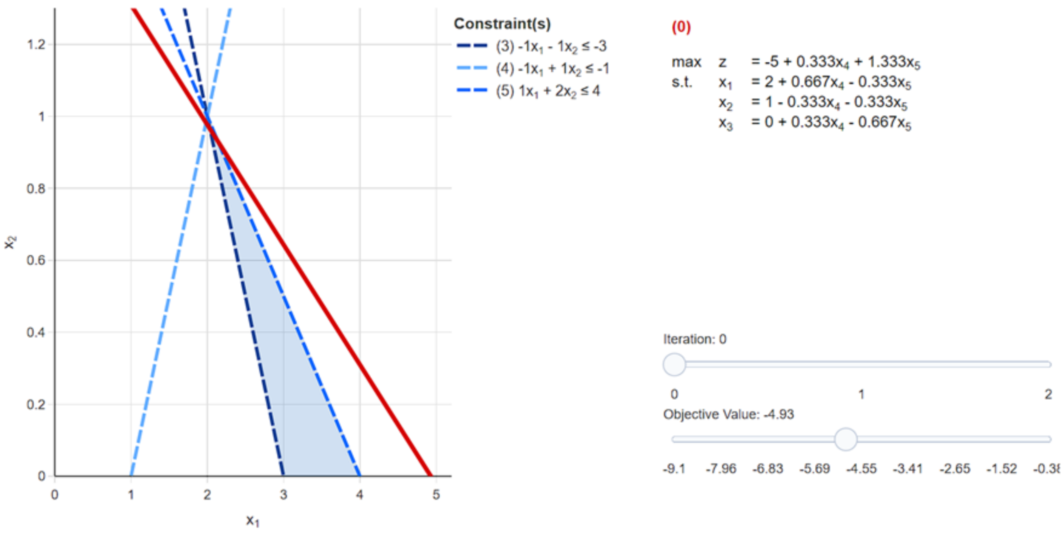
Ở giao diện này thì ta sẽ có đề bài, và kết quả. Kết quả bao gồm:

- Phương pháp giải: 2 pha
- Trạng thái: đã được giải
- Giá trị tối ưu z = -5
- Nghiệm tối ưu x1 = 2, x2= 1

Còn chức năng nữa dùng để vẽ hình trong cho hai biến nhấn nút show figure



Hình sẽ được mở trên browser. Trên hình ta có Objective values để di chuyển hàm mục tiêu

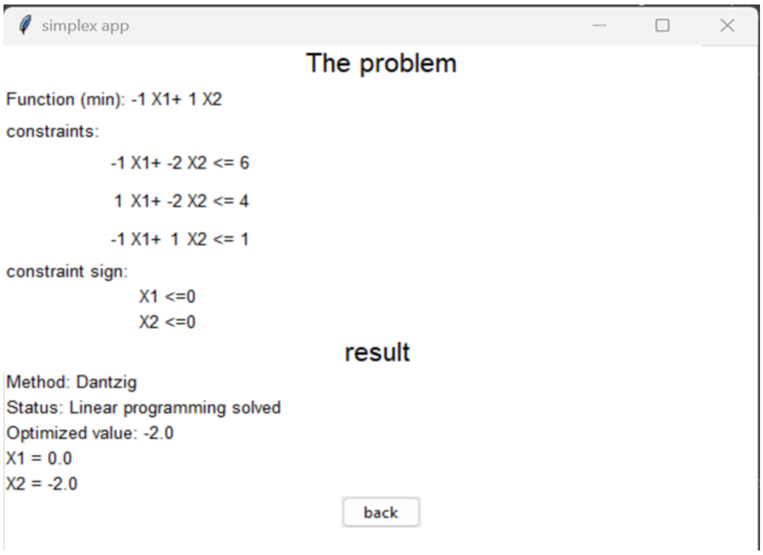


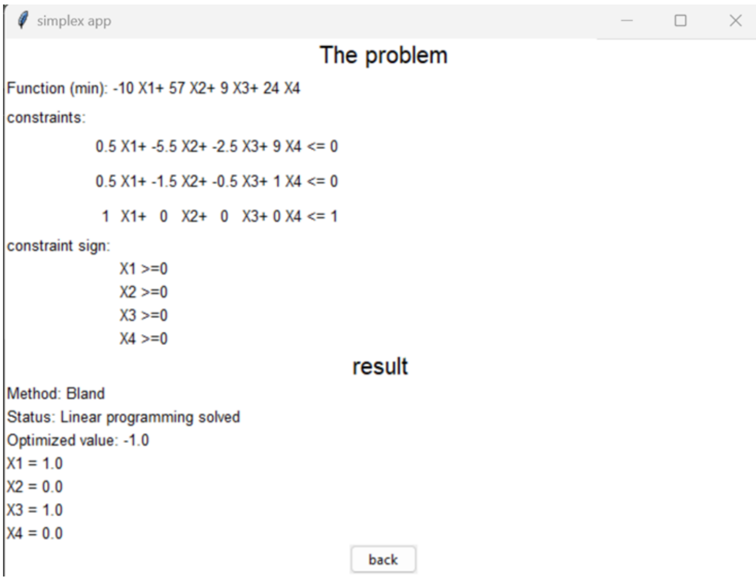
Sau khi giải xong ta có thể nhấn back để quay về ban đầu giải bài tiếp theo hoặc đóng chương trình.

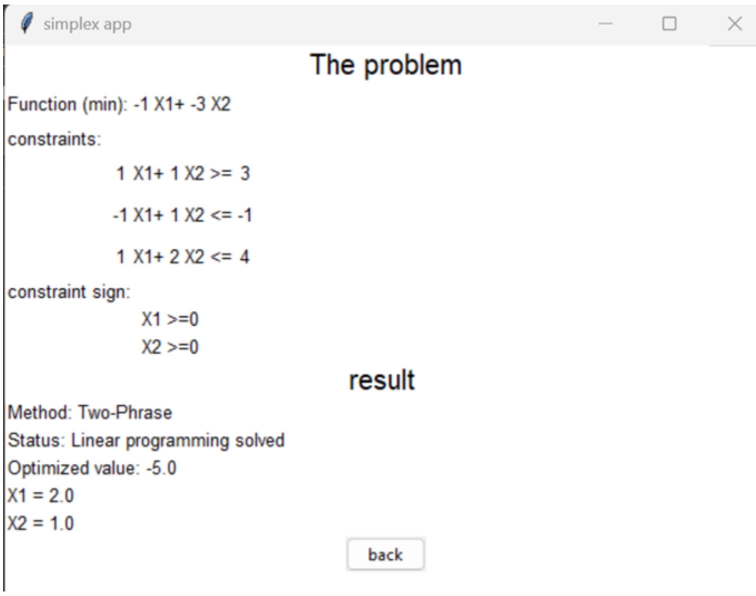
2 Test case thực tế

- Linear programming solved: trường hợp ta có đầy đủ nghiệm
- Linear programming infinite solution: trường số có vô nghiệm(hiện thị một số nghiệm)
- Linear programming unbounded: trường hợp không giới nội
- linear programming infeasible: trường hợp vô nghiệm

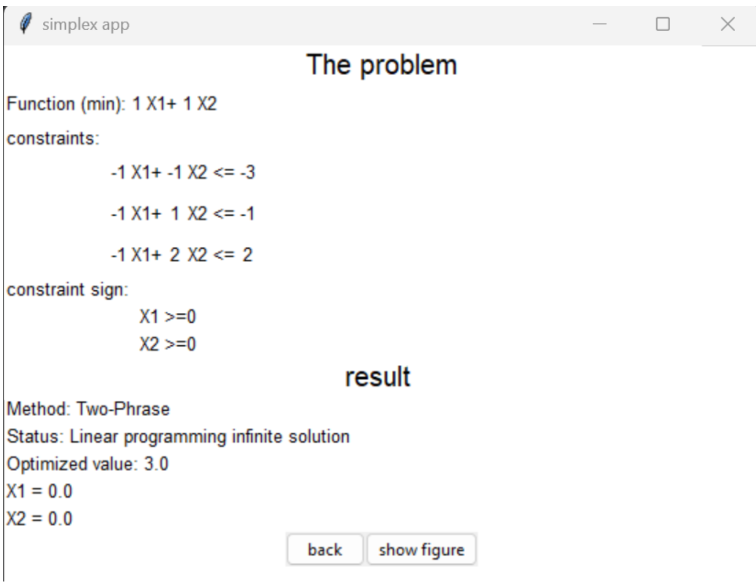
1. có nghiệm



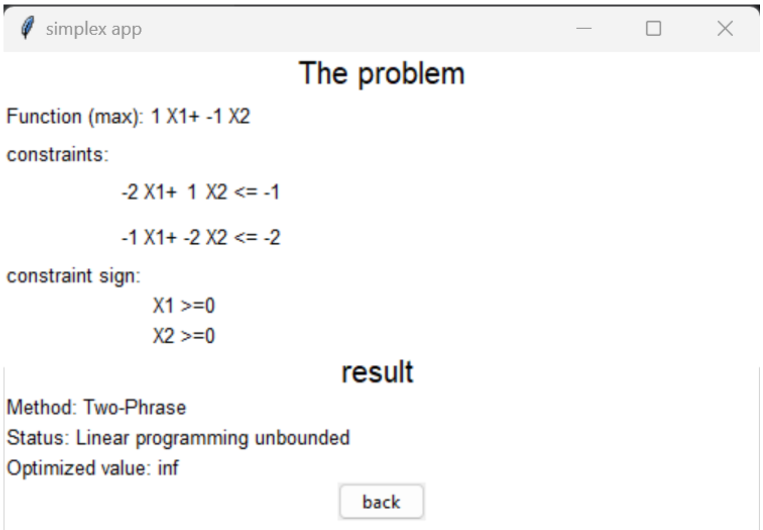


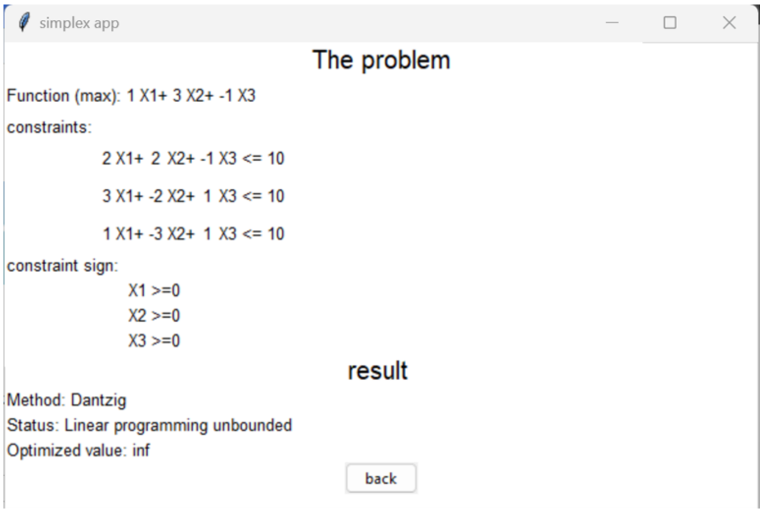


2. Vô số nghiệm

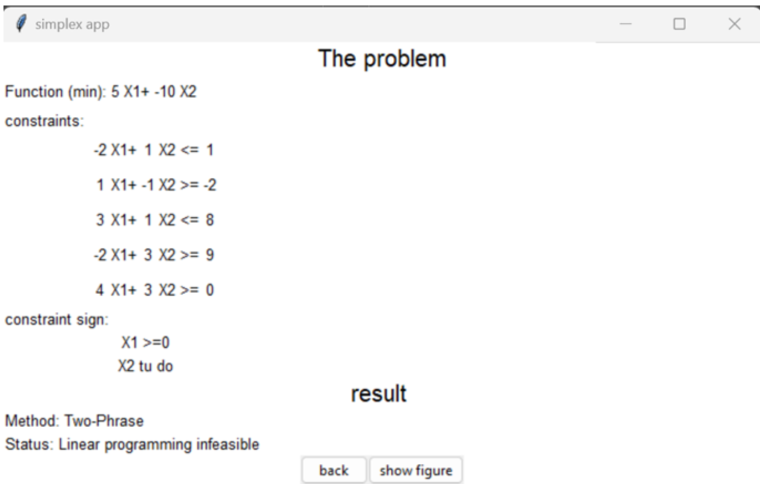


3. Không giới nội





4. Vô nghiệm



3 Cách thuật toán hoạt động:

3.1 Cấu trúc dữ liệu bảng

Tổ chức trong lớp Tableau, với các hệ số của ràng buộc là ma trận $A(nxm)$, hệ số hàm mục tiêu vector c và hệ số tự do của từng ràng buộc b. Ngoài 3 đầu vào trên lưu lại các ràng buộc dấu cho biến và ràng buộc dấu cho đẳng thức. Khi đó:

$$A_{m\times n} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix}, b_{m\times 1} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{bmatrix}, c^T_{1\times n} = \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_m \end{bmatrix}$$

Biến đổi thành dạng:

X	W	Rhs
$A_{m\times n}$	I_m	$b_{m\times 1}$
$c_{1\times n}$	[0, 0,..,0]	0

Với W là ma trận với các phần tử trên đường chéo chính là biến giả và vector 0 của các biến giả trên hàm mục tiêu, Rhs là vector hệ số tự do của các đẳng thức và giá trị mục tiêu ban đầu là 0.
Để đưa mọi bài toán quy hoạch tuyến tính về dạng chuẩn sau:

$$\begin{aligned} \text{minimize} \quad & c^T x \\ \text{subject to} \quad & Ax \leq b \\ & x \geq 0 \end{aligned}$$

Ta cần quan tâm các vấn đề sau:

1. Min, Max hàm mục tiêu
   - Với trường hợp Max: các hệ số hàm mục tiêu * -1. Ngược lại giữ nguyên hàm mục tiêu
2. Dấu của đẳng thức rằng buộc
   - Với trường hợp $Ax \geq b$: nhân mọi hàng (ràng buộc) xuất hiện $\geq$ cho -1. Hay ta nhân hệ số A_i và b_i cho -1 để đẳng thức sang dấu $\leq$
   - Với trường hợp $Ax = b$: ta tách phương trình thành hai đẳng thức con mà vẫn đảm bảo ràng buộc thõa mãn với dạng chuẩn

     $$Ax = b \Rightarrow \begin{cases} Ax \leq b \\ -Ax \leq -b \end{cases}$$

   - Trường hợp còn lại $Ax \leq b$ không thay đổi.
3. Dấu của mỗi biến:
   - Với $x \leq 0$: nhân hệ số cho -1. Hay nhân cột của ma trận cả A và c tại vị trí i cho -1
   - Với x tự do: theo lí thuyết ta tách ra làm 2 biến con $x = x^+ - x^-$ với $x^+, x^- \geq 0$

   $$\begin{bmatrix} & \downarrow & & & \\ a_{11} & a_{12} & 1 & 0 & b_1 \\ a_{21} & a_{22} & 0 & 1 & b_2 \\ c_1 & c_2 & 0 & 0 & 0 \end{bmatrix} \Rightarrow \begin{bmatrix} & \downarrow & \downarrow & & & \\ a_{11} & a_{12} & -a_{12} & 1 & 0 & b_1 \\ a_{21} & a_{22} & -a_{22} & 0 & 1 & b_2 \\ c_1 & c_2 & -c_2 & 0 & 0 & 0 \end{bmatrix}$$

4. Với kết quả nhận được khi thuật toán kết thúc trên từ vựng
   (a) Bài toán được gọi là tối ưu và có nghiệm duy nhất khi các hệ số hàm mục tiêu là dương. VD:

   $$\begin{bmatrix} 1 & 0 & 1 & 0 & 10 \\ 2 & 1 & 6 & 1 & 3 \\ 0 & 0 & 1 & 0 & 12 \end{bmatrix}$$

   (b) Bài toán được gọi là vố số nghiệm khi xuất hiện biến tuỳ ý mà không ảnh hưởng đến sự thay đổi của hàm mục tiêu. VD:

   $$\begin{bmatrix} 1 & 0 & 1 & 5 & 10 \\ 0 & 1 & 6 & 1 & 3 \\ 0 & 0 & 1 & 4 & 12 \end{bmatrix}$$

(c) Bài toán được gọi là không giới nội khi giá trị mục tiêu ∞

$$\begin{bmatrix} 1 & 0 & 1 & -5 & 10 \\ 0 & 1 & 6 & -1 & 3 \\ 0 & 0 & 1 & -4 & 12 \end{bmatrix}$$

(d) Bài toán được gọi là vô nghiệm khi tại bước k xử lí theo pha 1 (một phần thuật toán 2 pha) khi cho $x_0 = 0$ từ vựng không tối ưu hay không mang giá trị tối ưu là 0. Khi này các ràng buộc và hàm mục tiêu chịu sự thay đổi từ x_0

$$\begin{array}{cccccc} & & & & x_0 & \\ & & & & \downarrow & \end{array}$$
$$\begin{bmatrix} 1 & 0 & 1 & -5 & -1 & 10 \\ 0 & 1 & 6 & -1 & 2 & 3 \\ 0 & 0 & 1 & 4 & 3 & 12 \end{bmatrix}$$

3.2 Thuật toán Dantzig và Bland

Khởi tạo thuật toán với việc lưu bảng từ vựng được xây dựng, chỉ số của biến trên cột và trạng thái từ vựng ban đầu để so sánh khi từ vựng ở bước n bị lặp lại ban đầu để tránh lặp vô hạn. Thực hiện thuật toán thông qua **solve**. Mỗi lần lặp thuật toán kiểm tra có sự xuất hiện hệ số âm hàm mục tiêu hay không. Nếu có:

- Bài toán xem xét với $\forall b_i \geq 0$ thực hiện đơn hình Dantzig. Bằng cách chọn hệ số âm nhỏ nhất của hàm mục tiêu là biến vào (trên cột)
- Nếu $\exists b_i = 0$ thực hiện đơn hình Bland. Bằng cách chọn chỉ số nhỏ nhất mà ở đó hệ số của chúng là âm.

Sau khi đã chọn đường biến vào. Sử dụng **find_pivot** tìm biến ra trên các đẳng thức rằng buộc bằng việc tính hệ số $\frac{b_i}{a_i}$ với i đại diện cho dòng i của table, mà tỉ số này là dương nhỏ nhất khi $a_i > 0$ và $b_i \geq 0$. Đặc biệt sẽ có trường hợp có k tỉ số bằng nhau, khi này chọn a_i nhỏ nhất thuận tiện cho tính toán (vì nếu chọn số quá lớn để chia, một số trường hợp như $\frac{1}{10^n}$ rất nhỏ máy tính sẽ lầm tưởng là giá trị 0).
Tìm được pivot tiến hành xoay từ vựng bằng thuật toán Gauss_Jordan trừ các dòng khác i cho dòng i được nhân với hệ số dòng được trừ để triệt tiêu hệ số và kết quả tồn tại 1 pivot trên cột (ta có được biến vào)
Ta thực thi thuật toán Dantzig đến khi đạt từ vựng tối ưu với 3 kết qủa xảy ra như đã kể trên: Không giới nội khi không tìm được biến ra với tỉ lệ $\frac{b_i}{a_i}$ là những số âm, vô số nghiệm khi một biến không ảnh hưởng đến sự thay đổi của hàm mục tiêu, cuối cùng nghiệm duy nhất.

3.3 Thuật toán đơn hình 2 pha

Tương tự như 2 thuật toán trên, ta vẫn lưu lại bảng từ vựng được xây dựng trước, chỉ số của biến trên cột và hệ số mục tiêu ban đầu khi chuyển từ pha 1 sang pha 2.
Mục đích của 2 pha nhằm triệt tiêu hệ số tự do $b_i < 0$ của ràng buộc đẳng thức và tiến hành thực hiện thuật toán Dantzig hoặc Bland trên từ vựng mới với $b_i \geq 0$. Ở đây ta chỉ tập trung giải thích cách biến đổi pha 1 và chuyển sang bảng pha 2 còn thuật toán tìm nghiệm tối ưu cho pha 2 tương tự 2 thuật toán kể trên.
Pha 1, tạo bảng bằng việc thêm một cột biến giả trước cột hệ số b_i và đổi toàn bộ hệ số hàm mục tiêu thành 0 qua phương thức **phase1_tableau** như sau:

$$\begin{bmatrix} 1 & 0 & 1 & -5 & 10 \\ 0 & 1 & 6 & -1 & -3 \\ 3 & 2 & 0 & 0 & 0 \end{bmatrix} \Rightarrow \begin{array}{c} \begin{array}{cccccc} & & & & x_0 & \\ & & & & \downarrow & \end{array} \\ \begin{bmatrix} 1 & 0 & 1 & -5 & -1 & 10 \\ 0 & 1 & 6 & -1 & -1 & -3 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \end{array}$$

Tương tư như việc chuyển đổi bài toán dạng chuẩn:

$$\begin{array}{lllll} \text{minimize} & c^T x & & \text{minimize} & x_0 \\ \text{subject to} & Ax \leq b & \text{to} & \text{subject to} & Ax - x_0 \leq b \\ & x \geq 0 & & & x \geq 0, x_0 \geq 0 \end{array}$$

Thực thi thuật toán qua **dantzig_2Phase**. Điều kiện để kết thúc pha 1 khi hàm mục tiêu đạt được giá trị 0 khi chỉ tồn tại $x_0 = 0$ tại đó. Hay đẳng thức không phụ thuộc vào biến x_0. Giai đoạn tiếp theo chuyển sang pha 2. Lấy các biến bài toán đầu là biến cở sở ở pha 1 thay vào hàm mục tiêu ta được hàm mục tiêu mới. Lấy ví dụ từ bảng từ vựng trên. Biết rằng x_1, x_2 là biến cở sở khi đó hệ số chuyển toàn bộ hệ số biến không cở sở sang vế phải và thay vào hàm mục tiêu, được hiểu như sau:

$$\begin{array}{l} x_1 + w_1 - 5 * w_2 = 10 \\ \Leftrightarrow \quad x_1 = 10 - w_1 + 5 * w2 \end{array} \text{ và } \begin{array}{l} x_2 + 6 * w_1 - w_2 = 9 \\ \Leftrightarrow \quad x_1 = 9 - 6 * w_1 + w2 \end{array} \text{ và } \begin{array}{l} 3 * x_1 + 2 * x_2 \\ \Leftrightarrow \quad 48 - 15 * w_1 + 17 * w_2 \end{array}$$

Trên ma trận được biểu diễn với biến cở sở

$$3 * \begin{bmatrix} -1 \\ 5 \end{bmatrix} + 2 * \begin{bmatrix} -6 \\ 1 \end{bmatrix} = \begin{bmatrix} -15 \\ 17 \end{bmatrix}$$

Và hệ số tự do $3 * 10 + 2 * 9 = 48$. Sau khi có được hàm mục tiêu mới và ràng buộc từ pha 1 thực thi đơn hình Dantzig trên từ vựng xuất phát này đến khi tìm được nghiệm. Trong trường hợp khi pha 1 tối ưu với các hệ biến không cơ sở dương, ta nói bài toán vô nghiệm và không thực hiện pha 2.

3.4 Linear Program

Khởi tạo chương trình với bảng từ vựng, chỉ số của biến trên bảng, số ràng buộc đẳng thức, giá trị tối ưu, nghiệm tối ưu và phương pháp giải
__solve__ kiểm tra bảng từ vựng nếu hệ số $b < 0$ sử dụng giải thuật 2 pha, nếu $b > 0$ dùng Dantzig, trường hợp còn lại dùng Bland để giải quyết bài toán. Với mỗi thuật toán ta trả về từ vựng tối ưu và thông báo tình trạng bài toán.
__find__ Dựa vào thông báo của thuật toán trả về nghiệm tương ứng:

1. Không giới nội, nếu bài toán tìm max thì ∞ ngược lại $-\infty$
2. Vô nghiệm trả về nghiệm nan (hay không tồn tại nghiệm tối ưu)
3. Có nghiệm duy nhất: ứng với mỗi biến ở hàm mục tiêu quan sát trên bảng từ vựng nếu cột chỉ xuất hiện chỉ có giá trị 1 thì nghiệm tương ứng trên dòng đó, mặt khác nếu trên cột toàn hệ số khác 0 và không chỉ riêng 1 tức biến này là biến không cơ sở thì nghiệm của nó bằng 0.
4. Trường hợp vô sô nghiệm in ra giá trị tối ưu

4 Hạn chế

- Giao diện: chưa đạt được layout, kích thước cửa sổ phù hợp với từng bước khi thực hiện tính toán.
- Thuật toán chưa tối ưu khi chạy nhiều biến nhiều rằng buộc khi số biến và số ràng buộc lớn độ phức tạp tính toán và lưu trữ là $O(n^2)$.
- Trường hợp vô số nghiệm chưa in ra được nghiệm cơ sở phụ thuộc những biến nào.

5 Tài liệu tham khảo

- engri-1101/gilp
- tkinter documentation
- tkinter layout