# Programmazione Lineare

Di Giampietro Andrea, s5208458

## La richiesta

**Compito 3.1.** *Campionamento manuale*
Nel primo quadrante del piano cartesiano $X \times Y$ è dato il problema di programmazione lineare

$$\begin{array}{ll} \max & 2y - x \\ \text{con} & v_1 : \; y - x \leq 0 \\ & v_2 : \; y - 1 \leq 0 \\ & v_3 : \; y + x - 4 \leq 0 \end{array}$$

Disegna la regione ammissibile e il fascio improprio di rette parallele definito dalla funzione obiettivo. Risolvi il problema individuando la coppia di vincoli critici, il punto di massimo e la retta del fascio che lo contiene. Determina passo per passo come *LVIncrementalLP* risolve il problema se campioni

- $v_1$ da $\{v_1, v_2, v_3\}$ e $v_2$ da $\{v_2, v_3\}$ o
- $v_2$ da $\{v_1, v_2, v_3\}$ e $v_1$ da $\{v_1, v_3\}$

Ammetti come possibile il valore $+\infty$ nel caso in cui l'ascissa o l'ordinata del punto di massimo non è limitata.

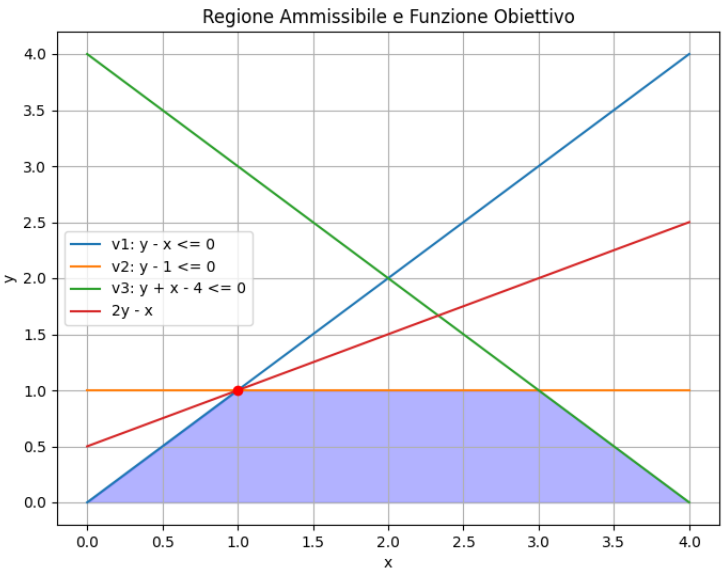
## L'approccio alla richiesta

Ho affrontato il problema della rappresentazione grafica della regione ammissibile sul piano cartesiano implementando un programma in Python. Questo mi ha permesso di ottenere una visione completa dei dati analizzati. Di seguito riporto le osservazioni principali.

## I dati ottenuti

Il programma sviluppato ha gestito i vincoli come equazioni lineari nel piano cartesiano, rappresentandoli come rette. L'intersezione tra esse ha permesso di determinare la regione ammissibile.

Confrontando le rette, si sono individuati i punti di intersezione che definiscono i limiti della regione ammissibile. Risolvendo i sistemi di equazioni lineari dati dai vincoli, sono stati ottenuti i punti di intersezione.



Successivamente, sostituendo le coordinate dei vertici della regione ammissibile nella funzione obiettivo, è stato determinato il punto di massimo, che in questo caso è risultato essere (1,1).

Analizzando graficamente i vincoli, è stato possibile osservare che la coppia di vincoli v1 e v2 è critica, poiché le rette corrispondenti passano per il punto di massimo.

## La seconda richiesta

Per poter analizzare il comportamento di LVIncrementalLP nei casi richiesti, ho ritenuto più efficace usare "carta e penna" per analizzare i casi passo per passo.

**Algoritmo 3.1.** *LVIncrementalLP*$(V)$
*Input*: un insieme $V$ di $|V|$ vincoli
*Output*: $\mathbf{x}^*$, ottimo per $V$, e $\mathcal{B}(V)$, base di $V$

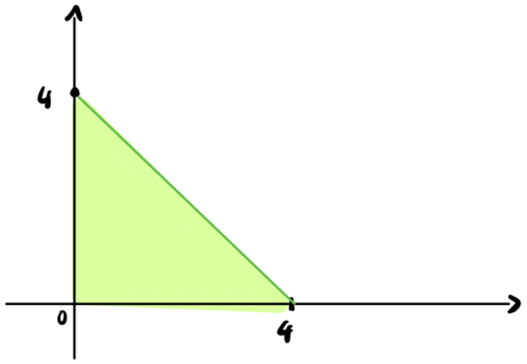
---

1. `if` $n = 1$ or $m = 1$
   determina $\mathbf{x}^*$
   `return` $\mathbf{x}^*$
2. campiona un vincolo $v$ uniformemente in $V$
3. *LVIncrementalLP*$(V \setminus \{v\})$
4. `if` $\mathbf{x}^*$ non viola $v$
   `return` $\mathbf{x}^*$
   `else`
   proietta i vincoli in $V \setminus \{v\}$ su $v$ ottenendo $V'$
   *LVIncrementalLP*$(V')$

### Come funziona l'algoritmo con v1, v2 e v3?

1) La prima verifica restituirà "false", quindi si procede con il campionamento del primo vincolo e si effettua una chiamata ricorsiva con i vincoli rimanenti non ancora considerati.
2) La condizione del passaggio 1 non viene nuovamente verificata. Si procede quindi al campionamento del secondo vincolo e si effettua un'altra chiamata ricorsiva con il vincolo rimanente.
3) La condizione del passaggio 1 viene verificata in questo punto, dato che rimane solo un vincolo (m = 1). Sostituendo i vertici della regione ammissibile corrente nella funzione obiettivo, si seleziona come punto di massimo quello che restituisce il valore più elevato. Successivamente, viene effettuata una nuova chiamata ricorsiva.
4) Nella chiamata successiva, si verifica la condizione del passaggio 4 per verificare se il punto di massimo attuale violi o meno il nuovo vincolo. Se il vincolo non viene violato, il punto di massimo attuale viene restituito nuovamente. In caso contrario, si effettua una chiamata ricorsiva sovrapponendo i vincoli precedentemente visti.
5) Verranno eseguite chiamate ricorsive sovrapponendo tutti i vincoli, finché presenti, per infine ottenere la regione del piano ammissibile con relativo punto di massimo.

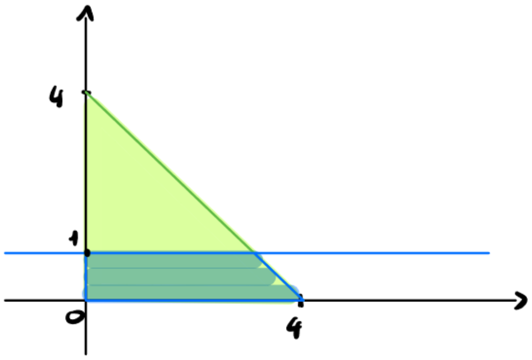
$v_1$ da $\{v_1, v_2, v_3\}$ e $v_2$ da $\{v_2, v_3\}$




Vincolo aggiunto al piano: v3

Vertici disponibili: (0, 4), (4, 0)

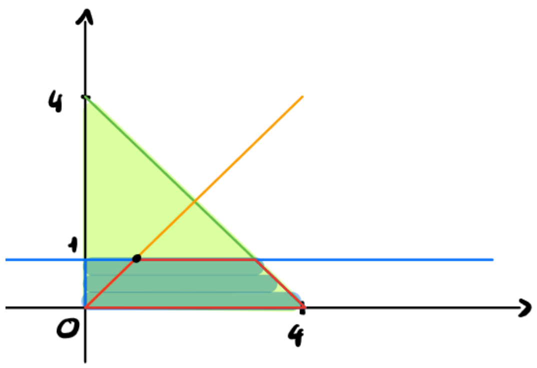
Punto di massimo attuale: (0, 4)




Vincolo aggiunto al piano: v2

Vertici disponibili: (0, 0), (0, 1), (3, 1), (4, 0)

Punto di massimo attuale: (0, 1)

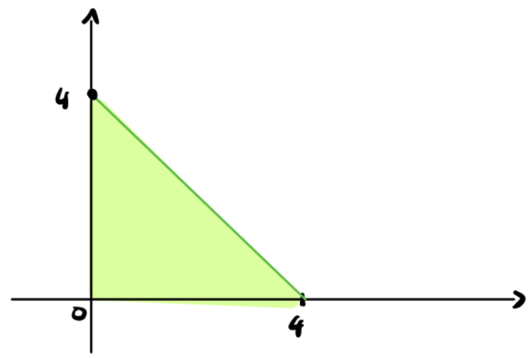



Vincolo aggiunto al piano: v3

Vertici disponibili: (0, 0), (1, 1),(3, 1), (4, 0)

Punto di massimo: (1, 1)

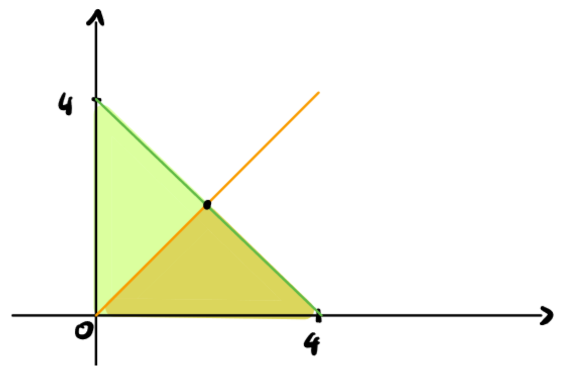
$v_2$ da $\{v_1, v_2, v_3\}$ e $v_1$ da $\{v_1, v_3\}$




Vincolo aggiunto al piano: v3

Vertici disponibili: (0, 4), (4, 0)

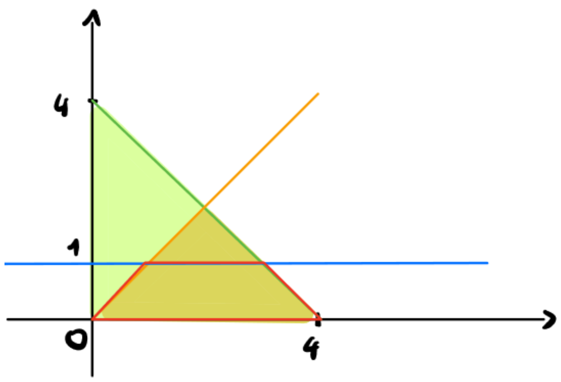
Punto di massimo attuale: (0, 4)




Vincolo aggiunto al piano: v1

Vertici disponibili: (0, 0), (2, 2), (4, 0)

Punto di massimo attuale: (2, 2)




Vincolo aggiunto al piano: v2

Vertici disponibili: (0, 0), (1, 1),(3, 1), (4, 0)

Punto di massimo: (1, 1)

## Il codice

```
#max -> 2y - x = 1 dopo aver trovato il vertice corrispondente al valore massimo della funzione
#v1 -> y - x <= 0
```

```python
#v2 -> y - 1 <= 0
#v3 -> y + x - 4 <= 0

import numpy as np
import matplotlib.pyplot as plt

# Definizione dei vincoli
v1 = lambda x: x
v2 = lambda x: 1
v3 = lambda x: 4 - x
f = lambda x: x*0.5 + 0.5 #2y - x -> y = 0.5x

# Intervallo di valori per x
x = np.linspace(0, 4, 100)

# Calcolo dei valori dei vincoli sulla griglia
y1 = v1(x)
y2 = v2(x)
y3 = v3(x)

# Calcolo dei valori della funzione obiettivo sulla griglia
f_values = f(x)


#Vertici della regione ammissibile, osservati dopo aver disegnato il grafico
vx1 = [0, 0]
vx2 = [1, 1]
vx3 = [3, 1]
vx4 = [4, 0]

# Calcolo il valore della funzione in ogni vertice della regione ammissibile
#formula: f(x) = 2y - x
f_vx1 = 2*vx1[1] - vx1[0]
f_vx2 = 2*vx2[1] - vx2[0]
f_vx3 = 2*vx3[1] - vx3[0]
f_vx4 = 2*vx4[1] - vx4[0]

# Determino il valore massimo della funzione e il vertice corrispondente
max = f_vx1
if f_vx2 > max:
    max = f_vx2
    max_vx = vx2
if f_vx3 > max:
    max = f_vx3
    max_vx = vx3
if f_vx4 > max:
    max = f_vx4
    max_vx = vx4
```

```python
# Stampo il vertice corrispondente al valore massimo della funzione
print("Il punto massimo della regione ammissibile è:", max_vx)


# Rappresentazione grafica della regione ammissibile e delle rette dei vincoli
plt.figure(figsize=(8, 6))
plt.plot(x, y1, label="v1: y - x <= 0")
plt.plot(x, y2*np.ones_like(x), label="v2: y - 1 <= 0")
plt.plot(x, y3, label="v3: y + x - 4 <= 0")
plt.fill_between(x, np.minimum(np.minimum(y1, y2), y3), where=(x >= 0) & (x <= 4), color='blue', alpha=0.3)
plt.plot(x, f_values, label="2y - x")

# Etichette degli assi
plt.xlabel("x")
plt.ylabel("y")

# Titolo del grafico
plt.title("Regione Ammissibile e Funzione Obiettivo")

# Mostra la legenda
plt.legend()

#Mostro il punto massimo
plt.plot(max_vx[0], max_vx[1], 'ro')


# Mostra il grafico
plt.grid(True)
plt.show()
```