Trioli Davide – S5316731 a.a. 2022/2023

Analisi e Progettazione di Algoritmi

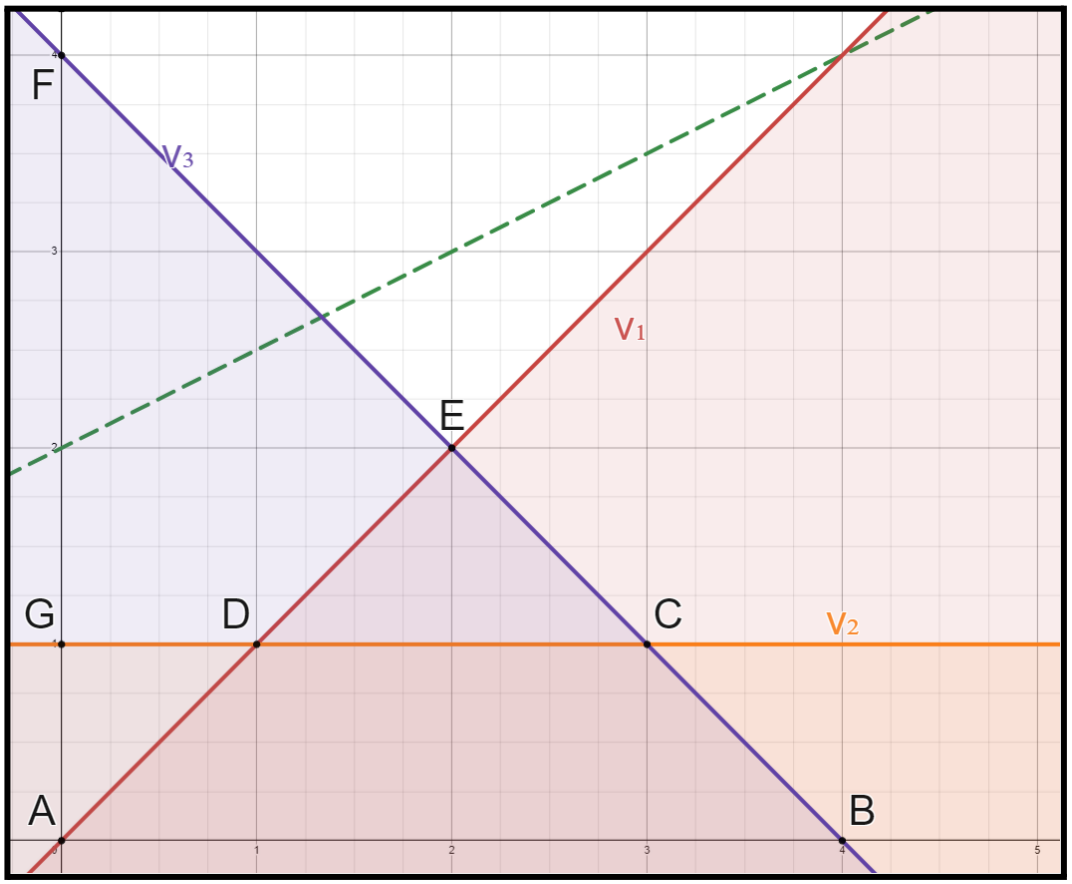
Esercizio 3.1
Lontani dal caso peggiore: Programmazione Lineare

Parte 1. *Geometricamente*

Il problema di programmazione lineare proposto prevede l'esistenza di una funzione obiettivo da massimizzare, $O(P)$ per un punto P, e di tre vincoli, v_1, v_2 e v_3.

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

Graficamente la situazione si può così rappresentare (qui disegnata attraverso *Desmos*):



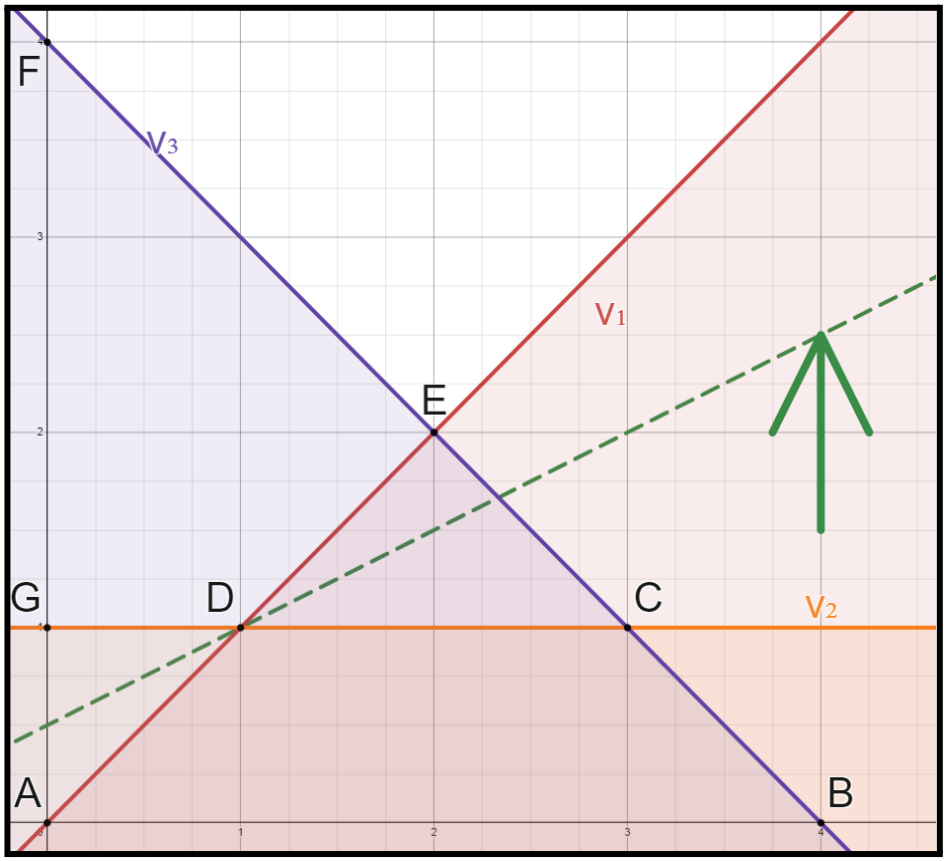
Dove i tre semipiani colorati sono generati dai tre vincoli indicati, mentre la retta tratteggiata verde è una qualsiasi rappresentante del fascio improprio delle rette parallele definite dalla funzione obiettivo; sono anche stati assegnati nomi ai punti di intersezione delle rette delimitatrici dei semipiani dei vincoli per facilitare la presentazione della parte successiva.
Da come si può notare anche cromaticamente la regione ammissibile così prodotta è quella più densamente colorata, ossia l'intersezione limitata solo al primo quadrante tra i semipiani vincolanti, corrispondente al quadrilatero $ABCD$.

Per individuare il punto di massimo geometricamente è necessario come primo passo calcolare le coordinate dei punti della regione ammissibile, intersecando le due rette generatrici di ognuno di essi:

- A: $\{ y - x = 0,\ y = 0 \} \Rightarrow \{ x = 0,\ y = 0 \} \Rightarrow A(0,\ 0)$
- B: $\{ y + x - 4 = 0,\ y = 0 \} \Rightarrow \{ x - 4 = 0,\ y = 0 \} \Rightarrow B(4,\ 0)$
- C: $\{ y + x - 4 = 0,\ y - 1 = 0 \} \Rightarrow \{ y = 1,\ 1 + x - 4 = 0 \} \Rightarrow C(3,\ 1)$
- D: $\{ y - x = 0,\ y - 1 = 0 \} \Rightarrow \{ y = 1,\ 1 - x = 0 \} \Rightarrow D(1,\ 1)$

Una volta ottenute le informazioni spaziali, si può facilmente individuare il punto di massimo sostituendo le coordinate di ogni punto alla quantità da massimizzare:

- $O(A) = 2(0) - 0 = 0$
- $O(B) = 2(0) - 4 = -4$
- $O(C) = 2(1) - 3 = 0$
- $O(D) = 2(1) - 1 = 1$



Si scopre quindi che il punto di massimo è D: in effetti, immaginando di far traslare sull'asse y la retta rappresentante della funzione obiettivo, l'ultimo punto più in alto toccato prima di allontanarsi dalla regione ammissibile è proprio D; si ha quindi che la coppia di vincoli critici è composta da v_1 e v_2.

Conoscendo il punto toccato e calcolando l'inclinazione della retta m, si può anche individuare esplicitamente la retta in questione:

$O(x) = 2y - x \Leftrightarrow y = \frac{O(x)+x}{2} \Leftrightarrow y = \frac{1}{2}x + \frac{1}{2}O(x) \Rightarrow m = \frac{1}{2}$

Applicando i dati conosciuti alla formula generale si ha

$y - y_D = m\left(x - x_D\right) \rightsquigarrow y - 1 = \frac{1}{2}(x - 1) \Rightarrow y = \frac{1}{2}x + \frac{1}{2}$.

Parte 2. *Algoritmicamente*

Sfruttando lo pseudocodice dell'algoritmo proposto `LVIncrementalLP`, si può indicare come lo stesso problema verrebbe risolto "immaginando" diverse configurazioni che in realtà avverrebbero in maniera casuale.

La prima situazione descritta prevede il seguente ordine dei campionamenti:

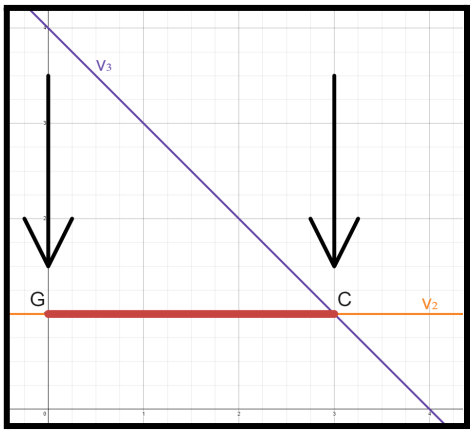
v_1 da $\{v_1, v_2, v_3\}$ e v_2 da $\{v_2, v_3\}$

(nell'interpretazione seguente si indica con una notazione insiemistica il set di vincoli che vengono inseriti in input per ogni chiamata a funzione, in modo da poter illustrare meglio cosa stia accadendo all'interno di ognuna di esse)

`LVIncrementalLP`$(\{v_1, v_2, v_3\})$:

- Inizio con 3 vincoli e 2 dimensioni, skip ramo if
- Campionamento v_1
- Chiamata `LVIncrementalLP`$(\{v_2, v_3\})$:
  - Inizio con 2 vincoli e 2 dimensioni, skip ramo if
  - Campionamento v_2
  - Chiamata `LVIncrementalLP`$(\{v_3\})$:
    - Inizio con 1 vincolo e 2 dimensioni, entrata ramo if
    - Calcolo ottimo locale $x*$ nel punto F, restituzione $x* = F$
  - $x* = F$ viola v_2:
  - Proiezione v_3 su v_2 genera segmento $\overline{GC}$ (fig. d'esempio a fondo pagina)
  - Chiamata `LVIncrementalLP`$(\overline{GC})$:
    - Inizio con 1 vincolo e 1 dimensione, entrata ramo if
    - Calcolo ottimo locale $x*$ nel punto G, restituzione $x* = G$
- $x* = G$ viola v_1:
- Proiezione v_2, v_3 su v_1 genera segmento $\overline{AD}$
- Chiamata `LVIncrementalLP`$(\overline{AD})$:
  - Inizio con 1 vincolo e 1 dimensione, entrata ramo if
  - Calcolo ottimo locale $x*$ nel punto D, restituzione $x* = D$

Fine algoritmo: dopo diverse chiamate ricorsive, il risultato ottenuto è quello corretto D.



sopra: proiezione di v_3 su v_2

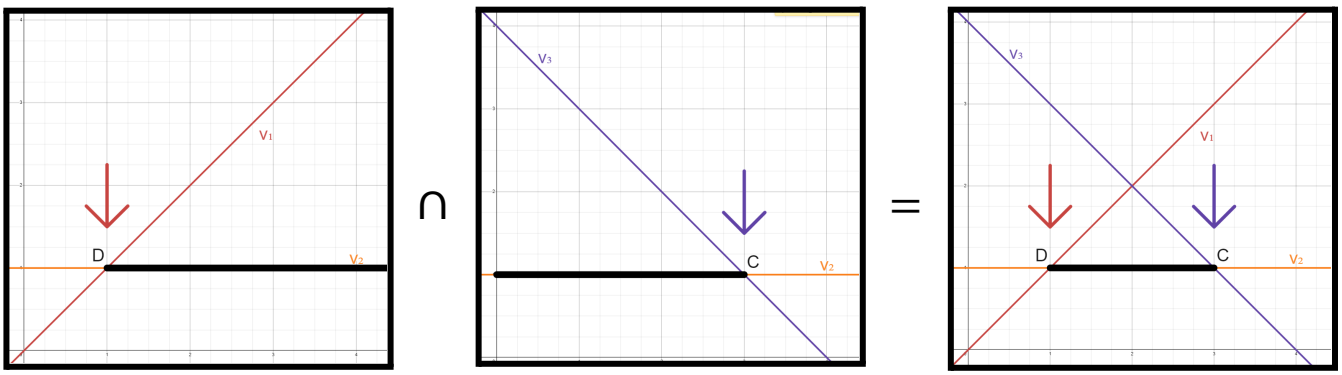
La seconda situazione descritta prevede i seguenti campionamenti:

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

`LVIncrementalLP`$(\{v_1, v_2, v_3\})$:

- Inizio con 3 vincoli e 2 dimensioni, skip ramo if
- Campionamento v_2
- Chiamata `LVIncrementalLP`$(\{v_1, v_3\})$:
  - Inizio con 2 vincoli e 2 dimensioni, skip ramo if
  - Campionamento v_1
  - Chiamata `LVIncrementalLP`$(\{v_3\})$:
    - Inizio con 1 vincolo e 2 dimensioni, entrata ramo if
    - Calcolo ottimo locale $x*$ nel punto F, restituzione $x* = F$
  - $x* = F$ viola v_1:
  - Proiezione v_3 su v_1 genera segmento $\overline{AE}$
  - Chiamata `LVIncrementalLP`$(\overline{AE})$:
    - Inizio con 1 vincolo e 1 dimensione, entrata ramo if
    - Calcolo ottimo locale $x*$ nel punto E, restituzione $x* = E$
- $x* = E$ viola v_2:
- Proiezione v_1, v_3 su v_2 genera segmento $\overline{DC}$ (fig. d'esempio a fondo pagina)
- Chiamata `LVIncrementalLP`$(\overline{DC})$:
  - Inizio con 1 vincolo e 1 dimensione, entrata ramo if
  - Calcolo ottimo locale $x*$ nel punto D, restituzione $x* = D$

Fine algoritmo: analogamente a quanto ottenuto secondo la precedente configurazione, l'ottimo risulta essere locato in D.



sopra, nell'ordine: la proiezione di v_1 su v_2, la proiezione di v_3 su v_2 e la loro intersezione, ossia la proiezione di entrambi su v_2.