Algoritmi avansați

C12, C13 - Elemente de programare liniară

Mihai-Sorin Stupariu

Sem. al II-lea, 2023-2024

Motivație: turnarea pieselor în matrițe

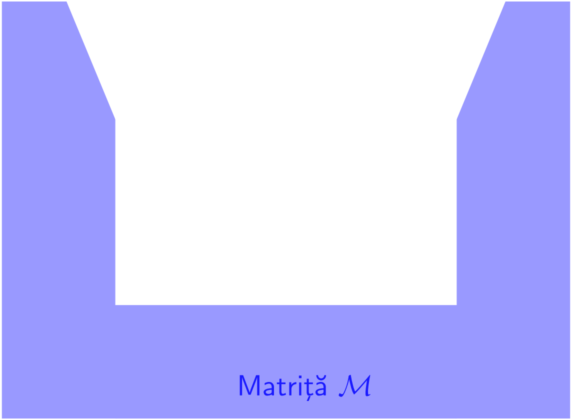
Intersecții de semiplane - abordare cantitativă

Dualitate

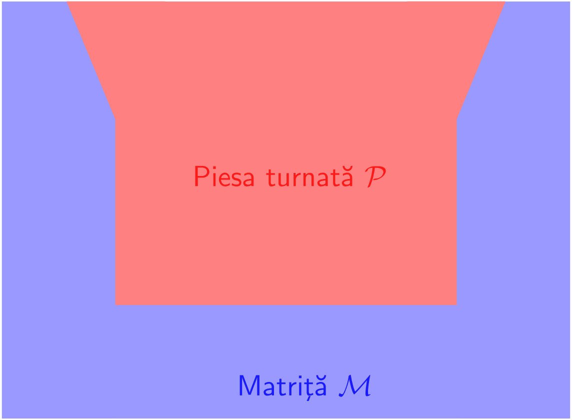
Intersecții de semiplane - abordare calitativă. Programare liniară

Subdiviziuni planare. DCEL

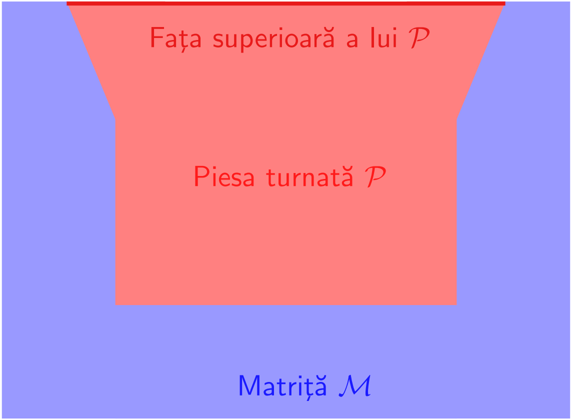
Turnarea pieselor în matrițe



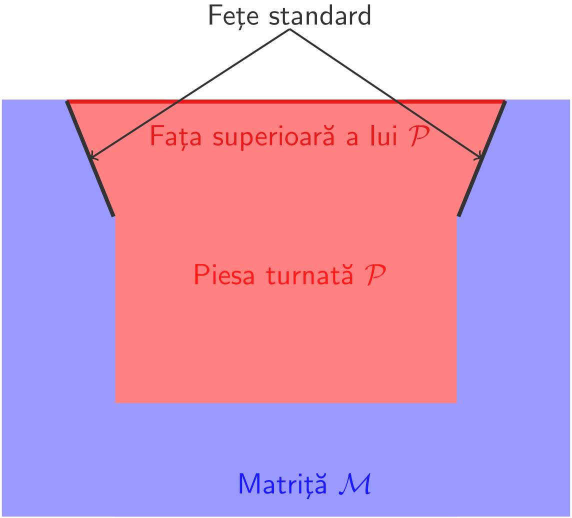
Turnarea pieselor în matrițe



Turnarea pieselor în matrițe



Turnarea pieselor în matriţe



Problematizare

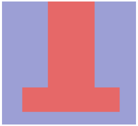
- Turnarea pieselor în matrițe și extragerea lor fără distrugerea matriței.

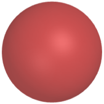
Problematizare

- Turnarea pieselor în matrițe și extragerea lor fără distrugerea matriței.
- Neajunsuri: unele obiecte pot rămâne blocate; există obiecte pentru care nu există o matriță adecvată.

Problematizare

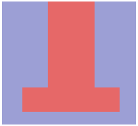
- Turnarea pieselor în matrițe și extragerea lor fără distrugerea matriței.
- Neajunsuri: unele obiecte pot rămâne blocate; există obiecte pentru care nu există o matriță adecvată.

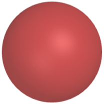




Problematizare

- Turnarea pieselor în matrițe și extragerea lor fără distrugerea matriței.
- Neajunsuri: unele obiecte pot rămâne blocate; există obiecte pentru care nu există o matriță adecvată.

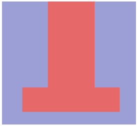


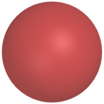


- **Problema studiată.** Dat un obiect, există o matriță din care să poată fi extras (și dacă da, cu un algoritm eficient?)

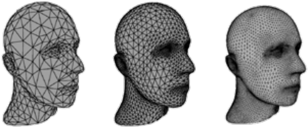
Problematizare

- Turnarea pieselor în matrițe și extragerea lor fără distrugerea matriței.
- Neajunsuri: unele obiecte pot rămâne blocate; există obiecte pentru care nu există o matriță adecvată.





- **Problema studiată.** Dat un obiect, există o matriță din care să poată fi extras (și dacă da, cu un algoritm eficient?)



Sursa: https://www.graphics.rwth-aachen.de/publication/03149/

Convenții

- Obiectele: **poliedrale**.

Convenții

- Obiectele: **poliedrale**.
- Matrițele: formate dintr-o singură piesă; fiecărui obiect $\mathcal{P}$ îi este asociată o matriță $\mathcal{M}_{\mathcal{P}}$

Convenții

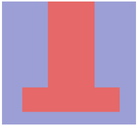
- Obiectele: **poliedrale**.
- Matrițele: formate dintr-o singură piesă; fiecărui obiect $\mathcal{P}$ îi este asociată o matriță $\mathcal{M}_{\mathcal{P}}$
- Obiectul: extras printr-o singură translație (sau o succesiune de translații)

Convenții

- Obiectele: **poliedrale**.
- Matrițele: formate dintr-o singură piesă; fiecărui obiect $\mathcal{P}$ îi este asociată o matriță $\mathcal{M}_{\mathcal{P}}$
- Obiectul: extras printr-o singură translație (sau o succesiune de translații)
- **Alegerea orientării:** diverse orientări ale obiectului pot genera diverse matrițe, astfel încât doar în unele configurații este posibilă extragerea obiectului

Convenții

- Obiectele: **poliedrale**.
- Matrițele: formate dintr-o singură piesă; fiecărui obiect $\mathcal{P}$ îi este asociată o matriță $\mathcal{M}_{\mathcal{P}}$
- Obiectul: extras printr-o singură translație (sau o succesiune de translații)
- **Alegerea orientării:** diverse orientări ale obiectului pot genera diverse matrițe, astfel încât doar în unele configurații este posibilă extragerea obiectului



Terminologie și convenții

- **Fața superioară:** prin convenție, obiectele au (cel puțin) o fața superioară (este orizontală, este singura care nu este adiacentă cu matrița). Celelalte fețe: **standard**; orice față standard f a obiectului corespunde unei fețe standard $\hat{f}$ a matriței.

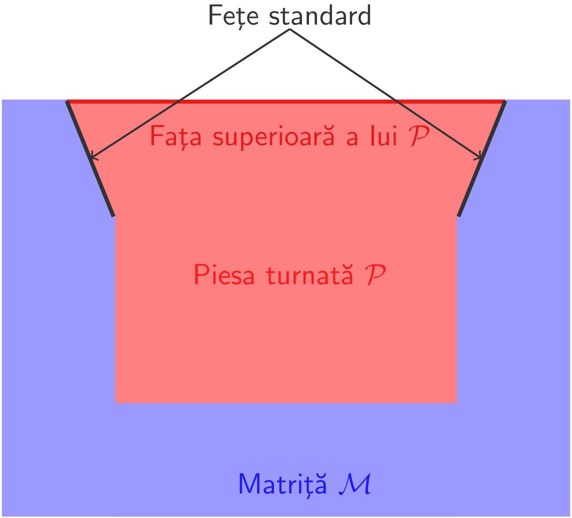
Terminologie și convenții

- **Fața superioară:** prin convenție, obiectele au (cel puțin) o fața superioară (este orizontală, este singura care nu este adiacentă cu matrița). Celelalte fețe: **standard**; orice față standard f a obiectului corespunde unei fețe standard $\hat{f}$ a matriței.
- **Obiect care poate fi turnat (*castable*):** există o orientare pentru care acesta poate fi turnat și apoi extras printr-o translație (succesiune de translații): *direcție admisibilă*.

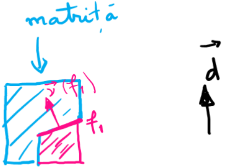
Terminologie și convenții

- **Fața superioară:** prin convenție, obiectele au (cel puțin) o fața superioară (este orizontală, este singura care nu este adiacentă cu matrița). Celelalte fețe: **standard**; orice față standard f a obiectului corespunde unei fețe standard $\hat{f}$ a matriței.
- **Obiect care poate fi turnat (*castable*):** există o orientare pentru care acesta poate fi turnat și apoi extras printr-o translație (succesiune de translații): *direcție admisibilă*.
- **Convenții:** Matrița este paralelipipedică și are o cavitate corespunzătoare obiectului; fața superioară a obiectului (și a matriței) este perpendiculară cu planul Oxy.

Turnarea pieselor în matrițe



Descrierea proprietății de a putea extrage o piesă într-o direcție dată



$\vec{\nu}(f_1)$ = normala exterioară la f_1

$\vec{\nu}(f_2)$

fața $\hat{f_1}$ a matriței blochează extragerea în direcția $\vec{d}$ <=>
$m(\sphericalangle(\vec{\nu}(f_1), \vec{d})) < 90^\circ$
<=> $\cos(\sphericalangle(\vec{\nu}(f_1), \vec{d})) > 0$

—"— $\hat{f_2}$ <u>nu</u> blochează
<=> $m(\sphericalangle(\vec{\nu}(f_2), \vec{d})) \geq 90^\circ$
<=> $\cos(\vec{\nu}(f_2), \vec{d}) \leq 0$

Această condiție tb. verificată ptr. toate fețele!

Detaliere (scriere în coordonate)

Dați $v, w \in \mathbb{R}^3$: $\cos(\sphericalangle(v, w)) = \frac{\langle v, w \rangle}{\|v\| \, \|w\|}$ $\left(\langle v, w \rangle = v_1 w_1 + v_2 w_2 + v_3 w_3 \right)$

Cum vrem să extragem obiectul "în sus", f.r.g. putem pp. că $\vec{d} = (d_x, d_y, 1)$ (de ce?)

Fie f o fața fixată a obiectului; $\vec{\nu}(f) = (\nu_x, \nu_y, \nu_z)$

Faptul că fața $\hat{f}$ a matriței nu blochează extragerea în direcția $\vec{d}$ $\Leftrightarrow$

$$\langle \vec{\nu}(f), \vec{d} \rangle \le 0 \Leftrightarrow$$

$$\boxed{\nu_x \cdot d_x + \nu_y \cdot d_y + \nu_z \le 0} \quad (*_f)$$

Fixată f ($\rightarrow (\nu_x, \nu_y, \nu_z)$) căutăm $\vec{d}\,(d_x, d_y)$ a.î. să fie verificată $(*_f)$

$\boxed{(*_f): \text{inecuație care descrie un semiplan}}$

Sinteză - fundamente geometrice

- **Condiție necesară:** direcția de extragere $\vec{d}$ trebuie să aibă componenta z pozitivă
- **În general:** o față standard $\hat{f}$ a matriței (corespunzătoare unei fețe f a piesei) pentru care unghiul dintre normala exterioară $\vec{\nu}(f)$ la fața f și $\vec{d}$ este mai mic de 90° împiedică translația în direcția $\vec{d}$

Sinteză - fundamente geometrice

- **Condiție necesară:** direcția de extragere $\vec{d}$ trebuie să aibă componenta z pozitivă
- **În general:** o față standard $\hat{f}$ a matriței (corespunzătoare unei fețe f a piesei) pentru care unghiul dintre normala exterioară $\vec{\nu}(f)$ la față f și $\vec{d}$ este mai mic de 90° împiedică translația în direcția $\vec{d}$
- **Propoziție.** *Un poliedru $\mathcal{P}$ poate fi extras din matrița sa $\mathcal{M}_{\mathcal{P}}$ prin translație în direcția $\vec{d}$ dacă și numai dacă $\vec{d}$ face un unghi de cel puțin 90° cu normala exterioară a fiecărei fețe standard a lui $\mathcal{P}$.*
- **Reformulare.** Dat $\mathcal{P}$, trebuie găsită o direcție $\vec{d}$ astfel încât, pentru fiecare față standard f, unghiul dintre $\vec{d}$ și $\vec{\nu}(f)$ să fie cel puțin 90°.

Sinteză - fundamente geometrice

- **Condiție necesară:** direcția de extragere $\vec{d}$ trebuie să aibă componenta z pozitivă
- **În general:** o față standard $\hat{f}$ a matriței (corespunzătoare unei fețe f a piesei) pentru care unghiul dintre normala exterioară $\vec{\nu}(f)$ la față f și $\vec{d}$ este mai mic de 90° împiedică translația în direcția $\vec{d}$
- **Propoziție.** *Un poliedru $\mathcal{P}$ poate fi extras din matrița sa $\mathcal{M}_{\mathcal{P}}$ prin translație în direcția $\vec{d}$ dacă și numai dacă $\vec{d}$ face un unghi de cel puțin 90° cu normala exterioară a fiecărei fețe standard a lui $\mathcal{P}$.*
- **Reformulare.** Dat $\mathcal{P}$, trebuie găsită o direcție $\vec{d}$ astfel încât, pentru fiecare față standard f, unghiul dintre $\vec{d}$ și $\vec{\nu}(f)$ să fie cel puțin 90°.
- **Analitic - pentru o față:** fiecare față definește un semiplan, i.e. dată o față standard f a poliedrului / matriței, a găsi o direcție admisibilă revine la a rezolva o inecuație $(*_f)$, care corespunde unui semiplan.

Sinteză - fundamente geometrice

- **Condiție necesară:** direcția de extragere $\vec{d}$ trebuie să aibă componenta z pozitivă
- **În general:** o față standard $\hat{f}$ a matriței (corespunzătoare unei fețe f a piesei) pentru care unghiul dintre normala exterioară $\vec{\nu}(f)$ la față f și $\vec{d}$ este mai mic de 90° împiedică translația în direcția $\vec{d}$
- **Propoziție.** *Un poliedru $\mathcal{P}$ poate fi extras din matrița sa $\mathcal{M}_{\mathcal{P}}$ prin translație în direcția $\vec{d}$ dacă și numai dacă $\vec{d}$ face un unghi de cel puțin 90° cu normala exterioară a fiecărei fețe standard a lui $\mathcal{P}$.*
- **Reformulare.** Dat $\mathcal{P}$, trebuie găsită o direcție $\vec{d}$ astfel încât, pentru fiecare față standard f, unghiul dintre $\vec{d}$ și $\vec{\nu}(f)$ să fie cel puțin 90°.
- **Analitic - pentru o față:** fiecare față definește un semiplan, i.e. dată o față standard f a poliedrului / matriței, a găsi o direcție admisibilă revine la a rezolva o inecuație $(*_f)$, care corespunde unui semiplan.
- **Analitic - toate fețele:** Fie $\mathcal{P}$ un poliedru; fața superioară fixată, paralelă cu planul Oxy. Considerăm matrița asociată și toate fețele matriței (i.e. toate fețele standard ale poliedrului). A determina o direcție admisibilă revine la a determina o direcție care verifică toate inegalitățile de tip $(*)$, deci un sistem de inecuații.

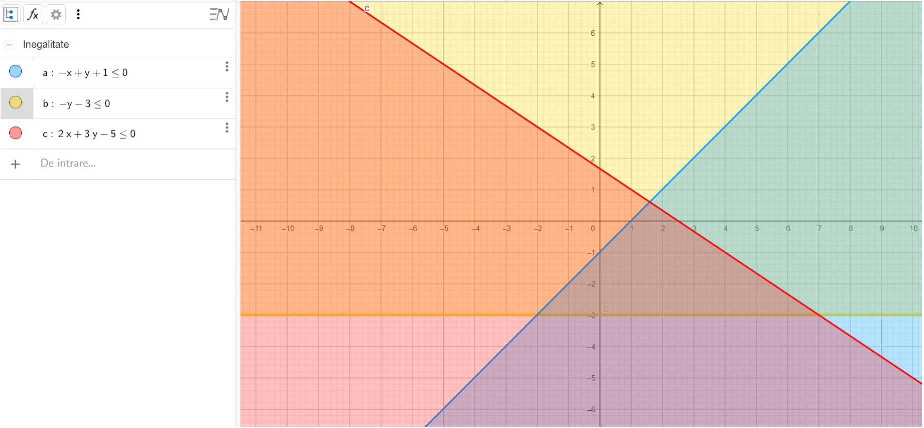
Sinteză - fundamente geometrice

- **Condiție necesară:** direcția de extragere $\vec{d}$ trebuie să aibă componenta z pozitivă
- **În general:** o față standard $\hat{f}$ a matriței (corespunzătoare unei fețe f a piesei) pentru care unghiul dintre normala exterioară $\vec{\nu}(f)$ la față f și $\vec{d}$ este mai mic de 90° împiedică translația în direcția $\vec{d}$
- **Propoziție.** *Un poliedru $\mathcal{P}$ poate fi extras din matrița sa $\mathcal{M}_\mathcal{P}$ prin translație în direcția $\vec{d}$ dacă și numai dacă $\vec{d}$ face un unghi de cel puțin 90° cu normala exterioară a fiecărei fețe standard a lui $\mathcal{P}$.*
- **Reformulare.** Dat $\mathcal{P}$, trebuie găsită o direcție $\vec{d}$ astfel încât, pentru fiecare față standard f, unghiul dintre $\vec{d}$ și $\vec{\nu}(f)$ să fie cel puțin 90°.
- **Analitic - pentru o față:** fiecare față definește un semiplan, i.e. dată o față standard f a poliedrului / matriței, a găsi o direcție admisibilă revine la a rezolva o inecuație $(*_f)$, care corespunde unui semiplan.
- **Analitic - toate fețele:** Fie $\mathcal{P}$ un poliedru; fața superioară fixată, paralelă cu planul Oxy. Considerăm matrița asociată și <u>toate</u> fețele matriței (i.e. toate fețele standard ale poliedrului). A determina o direcție admisibilă revine la a determina o direcție care verifică <u>toate</u> inegalitățile de tip $(*)$, deci un sistem de inecuații.
- **Concluzie:** Pentru a stabili dacă există o direcție admisibilă, trebuie stabilit dacă o <u>**intersecție de semiplane**</u> este nevidă.

Exemple

1. Intersecția semiplanelor

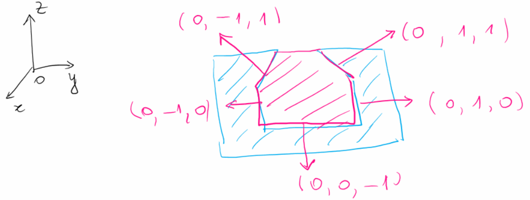
$$-x + y + 1 \leq 0; \quad -y - 3 \leq 0; \quad 2x + 3y - 5 \leq 0.$$



Exemple

2 (a). Normalele exterioare ale fețelor standard sunt coliniare cu vectorii

$$(0,-1,1),\ (0,1,1),\ (0,1,0),\ (0,0,-1),\ (0,-1,0).$$



sistem incompatibil, obiectul nu poate fi extras

$(0,-1,1) \rightsquigarrow 0\cdot x + (-1)\cdot y + 1 \le 0$ $y \ge 1$

$(0,1,1) \rightsquigarrow 0\cdot x + 1\cdot y + 1 \le 0$ $y \le -1$

$(0,1,0) \longrightarrow 0\cdot x + 1\cdot y + 0 \le 0$ $y \le 0$

$(0,0,-1) \longrightarrow 0\cdot x + 0\cdot y + (-1) \le 0$ $-1 \le 0$

$(0,-1,0) \rightsquigarrow 0\cdot x + (-1)\cdot y + 0 \le 0$ $y \ge 0$

Temă

2 (b). Normalele exterioare ale fețelor standard sunt coliniare cu vectorii

$$(0,1,0),\ (0,1,-1),\ (0,0,-1),\ (0,-1,-1),\ (0,-1,0).$$

Intersecții de semiplane - probleme studiate, rezultate

- **Probleme studiate:**

Intersecții de semiplane - probleme studiate, rezultate

- **Probleme studiate:**
  (i) **Caracterizare explicită:** Să se determine care sunt elementele (vârfuri, muchii, etc.) care determină o intersecție de semiplane.

Intersecții de semiplane - probleme studiate, rezultate

- **Probleme studiate:**
  (i) **Caracterizare explicită:** Să se determine care sunt elementele (vârfuri, muchii, etc.) care determină o intersecție de semiplane.
  (ii) **Calitativ:** Să se stabilească dacă o intersecție de semiplane este nevidă.

Intersecții de semiplane - probleme studiate, rezultate

- **Probleme studiate:**
  - (i) **Caracterizare explicită:** Să se determine care sunt elementele (vârfuri, muchii, etc.) care determină o intersecție de semiplane.
  - (ii) **Calitativ:** Să se stabilească dacă o intersecție de semiplane este nevidă.
- **Rezultate:** (descrise în detaliu ulterior)

Intersecții de semiplane - probleme studiate, rezultate

- **Probleme studiate:**
  - (i) **Caracterizare explicită:** Să se determine care sunt elementele (vârfuri, muchii, etc.) care determină o intersecție de semiplane.
  - (ii) **Calitativ:** Să se stabilească dacă o intersecție de semiplane este nevidă.
- **Rezultate:** (descrise în detaliu ulterior)
  - (i) *Intersecția unei mulțimi de n semiplane poate fi determinată cu complexitate-timp $O(n \log n)$ și folosind $O(n)$ memorie.*

Intersecții de semiplane - probleme studiate, rezultate

- **Probleme studiate:**
  (i) **Caracterizare explicită:** Să se determine care sunt elementele (vârfuri, muchii, etc.) care determină o intersecție de semiplane.
  (ii) **Calitativ:** Să se stabilească dacă o intersecție de semiplane este nevidă.
- **Rezultate:** (descrise în detaliu ulterior)
  (i) *Intersecția unei mulțimi de n semiplane poate fi determinată cu complexitate-timp $O(n \log n)$ și folosind $O(n)$ memorie.*
  (ii) *Se poate stabili cu complexitate-timp medie $O(n)$ dacă o intersecție de semiplane este nevidă.*
  (ii)' *Fie $\mathcal{P}$ un poliedru cu n fețe. Se poate decide dacă $\mathcal{P}$ reprezintă un obiect care poate fi turnat cu complexitate-timp medie $O(n^2)$ și folosind $O(n)$ spațiu. În caz afirmativ, o matriță și o direcție admisibilă în care poate fi extras $\mathcal{P}$ este determinată cu aceeași complexitate-timp.*

(i) Caracterizare explicită - Formularea problemei

- Fie $\mathcal{H} = \{H_1, H_2, \ldots, H_n\}$ o mulțime de semiplane din $\mathbb{R}^2$; semiplanul H_i dat de o relație de forma

$$a_i x + b_i y + c_i \leq 0$$

(i) Caracterizare explicită - Formularea problemei

- Fie $\mathcal{H} = \{H_1, H_2, \ldots, H_n\}$ o mulțime de semiplane din $\mathbb{R}^2$; semiplanul H_i dat de o relație de forma

$$a_i x + b_i y + c_i \leq 0$$

- Intersecția $H_1 \cap H_2 \cap \ldots \cap H_n$ este dată de un sistem de inecuații; este o mulțime poligonală convexă, mărginită de cel mult n muchii (poate fi vidă, mărginită, nemărginită,...)

(i) Caracterizare explicită - Formularea problemei

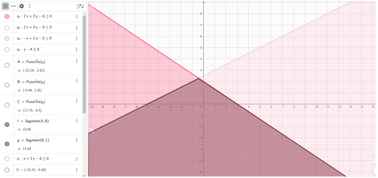
- Fie $\mathcal{H} = \{H_1, H_2, \ldots, H_n\}$ o mulțime de semiplane din $\mathbb{R}^2$; semiplanul H_i dat de o relație de forma

$$a_i x + b_i y + c_i \leq 0$$

- Intersecția $H_1 \cap H_2 \cap \ldots \cap H_n$ este dată de un sistem de inecuații; este o mulțime poligonală convexă, mărginită de cel mult n muchii (poate fi vidă, mărginită, nemărginită,...)
- De clarificat: date n semiplane, care sunt cele care contribuie **efectiv** la determinarea intersecției acestora? (v. slide-ul următor)

Semiplane - intersecții

Când determinăm o intersecție de semiplane, nu sunt neapărat relevante toate semiplanele. În figura de mai jos sunt considerate cinci semiplane s_1, s_2, s_3, s_4, s_5 dintre care relevante pentru intersecție sunt doar s_2 și s_4.



Dualitate – motivație intuitivă

- De câte informații (numerice) este nevoie pentru a indica **un punct în plan**?

Dualitate – motivație intuitivă

- De câte informații (numerice) este nevoie pentru a indica **un punct în plan**?
- **2**

Dualitate – motivație intuitivă

- De câte informații (numerice) este nevoie pentru a indica **un punct în plan**?
- **2**
- De câte informații (numerice) este nevoie pentru a indica **o dreaptă neverticală în plan**?

Dualitate – motivație intuitivă

- De câte informații (numerice) este nevoie pentru a indica **un punct în plan**?
- **2**
- De câte informații (numerice) este nevoie pentru a indica **o dreaptă neverticală în plan**?
- **2**

Dualitate – motivație intuitivă

- De câte informații (numerice) este nevoie pentru a indica **un punct în plan**?
- **2**
- De câte informații (numerice) este nevoie pentru a indica **o dreaptă neverticală în plan**?
- **2**
- Există o modalitate naturală de a stabili o corespondență între puncte și drepte?

Dualitate – motivație intuitivă

- De câte informații (numerice) este nevoie pentru a indica **un punct în plan**?
- **2**
- De câte informații (numerice) este nevoie pentru a indica **o dreaptă neverticală în plan**?
- **2**
- Există o modalitate naturală de a stabili o corespondență între puncte și drepte?
- **DA: dualitate**

Dualitate – motivație intuitivă

- De câte informații (numerice) este nevoie pentru a indica **un punct în plan**?
- **2**
- De câte informații (numerice) este nevoie pentru a indica **o dreaptă neverticală în plan**?
- **2**
- Există o modalitate naturală de a stabili o corespondență între puncte și drepte?
- **DA: dualitate**
- Cum se reflectă / respectă diferite proprietăți geometrice (de exemplu incidența) prin dualitate?

Dualitate – definiții

- Unui punct $p = (p_x, p_y)$ din planul $\mathbb{R}^2$ (plan primal) i se asociază o dreaptă din planul dual, notată p^*:

$$p^* : (y = p_x x - p_y) \text{ duala lui } p.$$

Dualitate – definiții

- Unui punct $p = (p_x, p_y)$ din planul $\mathbb{R}^2$ (plan primal) i se asociază o dreaptă din planul dual, notată p^*:

$$p^* : (y = p_x x - p_y) \text{ duala lui } p.$$

- Unei drepte neverticale $d : (y = m_d x + n_d)$ din planul primal i se asociază un punct din planul dual, notat d^*:

$$d^* = (m_d, -n_d) \text{ dualul lui } d.$$

Dualitate – definiții

- Unui punct $p = (p_x, p_y)$ din planul $\mathbb{R}^2$ (plan primal) i se asociază o dreaptă din planul dual, notată p^*:

$$p^* : (y = p_x x - p_y) \text{ duala lui } p.$$

- Unei drepte neverticale $d : (y = m_d x + n_d)$ din planul primal i se asociază un punct din planul dual, notat d^*:

$$d^* = (m_d, -n_d) \text{ dualul lui } d.$$

- **Obs.** Această transformare este polaritatea față de parabola $y = \frac{x^2}{2}$.

Dualitate – proprietăți elementare

1) Păstrează incidența

$p \in d \iff d^* \in p^*$

Exemplu

Pl. primal	Pl. dual
$d : (y = 2x + 1)$	$d^* = (2, -1)$
$p = (1, 3)$	$p^* : (y = x - 3)$

Dualitate – proprietăți elementare

2) Păstrează "ordinea"

p este situat deasupra dreptei d (neverticală) <=>

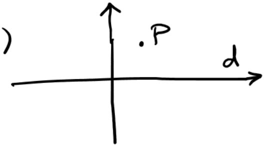
d^* — " — — " — p^*

Exemplu

Pl. primal

$p = (1,1)$

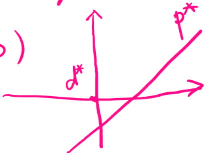
$d : (y = 0)$



Pl. dual

$p^* : (y = x - 1)$

$d^* = (0,0)$

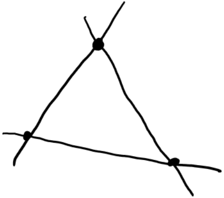


Dualitate – "dicționar" concepte și configurații

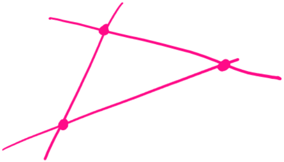
Plan primal	**Plan dual**
Punct p	Dreaptă neverticală p^*
Dreaptă neverticală d	Punct d^*
Dreaptă determinată de două puncte	Punct de intersecție a două drepte
Punctul p deasupra dreptei d	Punctul d^* deasupra dreptei p^*
Segment	Fascicul de drepte (*wedge*)

Exemplu

Configurația primală



Configurația duală



3 puncte necoliniare
și dreptele determinate
de ele

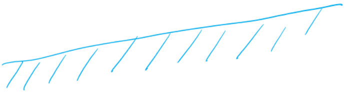
3 drepte care nu
trec prin același punct
și punctele determinate
de ele

Semiplane inferioare și semiplane superioare

- Exemple.

Semiplane inferioare și semiplane superioare

- Exemple.



semiplan inferior

semiplan superior

Semiplane inferioare și semiplane superioare

- Exemple.

semiplan inferior semiplan superior

- Dat un semiplan delimitat de o dreaptă neverticală

$$ax + by + c \leq 0$$

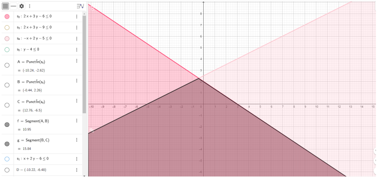
cum se decide dacă este semiplan inferior sau semiplan superior?
Exemple:

$$-x + y + 3 \leq 0 \qquad \text{semiplan inferior}$$

$$x - y - 3 \leq 0 \qquad \text{semiplan superior}$$

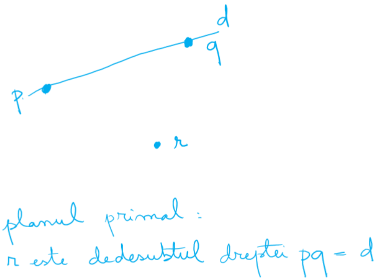
Semiplane inferioare și semiplane superioare

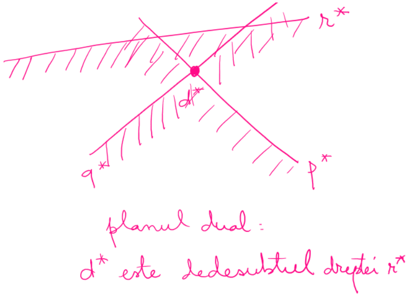
Când determinăm o intersecție de semiplane inferioare / superioare, nu sunt neapărat relevante toate semiplanele. În figura de mai jos sunt considerate cinci semiplane inferioare s_1, s_2, s_3, s_4, s_5 dintre care relevante pentru intersecție sunt doar s_2 și s_4.



Observația fundamentală

Fie p, q cu $p \neq q$ și dreapta $d = pq$ neverticală. Fie r un punct situat dedesubtul dreptei $d = pq$. Care este configurația duală?





Observația fundamentală

- Fie $\mathcal{P}$ o mulțime de puncte.

Observația fundamentală

- Fie $\mathcal{P}$ o mulțime de puncte.
- **Q:** Ce înseamnă că un segment $[pq]$ $(p, q \in \mathcal{P})$ participă la frontiera superioară a acoperirii convexe a lui $\mathcal{P}$?

Observația fundamentală

- Fie $\mathcal{P}$ o mulțime de puncte.
- **Q:** Ce înseamnă că un segment $[pq]$ $(p, q \in \mathcal{P})$ participă la frontiera superioară a acoperirii convexe a lui $\mathcal{P}$?
- **A:** Toate celelalte puncte sunt dedesubtul dreptei $d = pq$.

Observația fundamentală

- Fie $\mathcal{P}$ o mulțime de puncte.
- **Q:** Ce înseamnă că un segment $[pq]$ $(p, q \in \mathcal{P})$ participă la frontiera superioară a acoperirii convexe a lui $\mathcal{P}$?
- **A:** Toate celelalte puncte sunt dedesubtul dreptei $d = pq$.
- Configurația duală: Punctul d^* este situat dedesubtul dreptelor corespunzătoare celorlalte puncte și, prin trecere la semiplane inferioare, "contează" semiplanele inferioare determinate de p^* și q^*.

Concluzie pentru (i) - abordarea cantitativă

- Pentru a determina o intersecție de **semiplane inferioare** se consideră mulțimea de puncte din planul dual și se determină **frontiera superioară** a acoperirii convexe a mulțimii respective. Un rezultat analog are loc pentru intersecții de **semiplane superioar**e și **frontiera inferioară** a acoperirii convexe a mulțimii de puncte duale. În consecință:

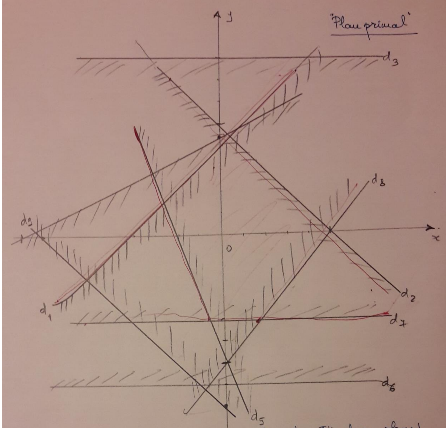
Concluzie pentru (i) - abordarea cantitativă

- Pentru a determina o intersecție de **semiplane inferioare** se consideră mulțimea de puncte din planul dual și se determină **frontiera superioară** a acoperirii convexe a mulțimii respective. Un rezultat analog are loc pentru intersecții de **semiplane superioar**e și **frontiera inferioară** a acoperirii convexe a mulțimii de puncte duale. În consecință:
- **Teoremă** *Intersecția a n semiplane poate fi descrisă cu un algoritm de complexitate* $O(n \log n)$.

Concluzie pentru (i) - abordarea cantitativă

- Pentru a determina o intersecție de **semiplane inferioare** se consideră mulțimea de puncte din planul dual și se determină **frontiera superioară** a acoperirii convexe a mulțimii respective. Un rezultat analog are loc pentru intersecții de **semiplane superioar**e și **frontiera inferioară** a acoperirii convexe a mulțimii de puncte duale. În consecință:
- **Teoremă** *Intersecția a n semiplane poate fi descrisă cu un algoritm de complexitate* $O(n \log n)$.
- De fapt, aplicarea dualității combinată cu sortarea lexicografică poate fi interpretată ca fiind ordonare a dreptelor după panta dreptei suport și apoi utilizarea unui mecanism incremental, cu eliminarea semiplanelor redundante. Această abordare directă este descrisă de Z. Zhu în 2006, în *New Algorithm for Half-plane Intersection and its Practical Value* (v. și această referință).

Exemplu



Configurația primală (intersecții de semiplane)

$\mathcal{LE}$: $y \leq x+4 \longrightarrow d_1: y = x+4$

$y \leq -x+5 \longrightarrow d_2: y = -x+5$

$y \leq 8 \longrightarrow d_3: y = 8$

$y \leq \frac{1}{2}x + 4.5 \rightarrow d_4: y = \frac{1}{2}x + 4.5$

$\mathcal{LE}$ e determinată de semiplanele asociate lui d_1 și d_2.

$\mathcal{UE}$: $y \geq -3x-6 \rightarrow d_5: y = -3x-6$

$y \geq -7 \longrightarrow d_6: y = -7$

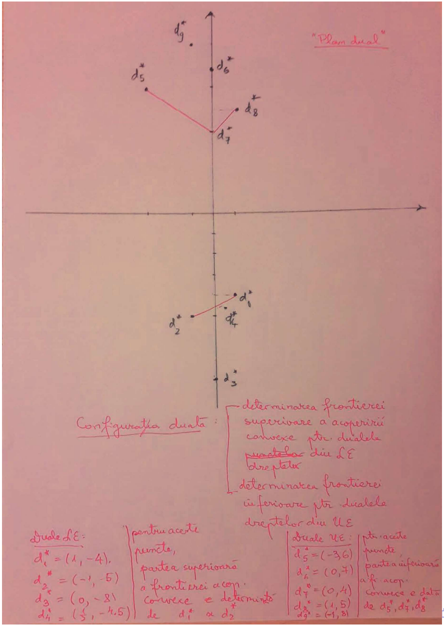
$y \geq -4 \longrightarrow d_7: y = -4$

$y \geq x-5 \rightarrow d_8: y = x-5$

$y \geq -x-8 \quad d_9: y = -x-8$

$\mathcal{UE}$ e determinată de semipl. asociate lui d_5, d_7, d_8.

Exemplu


"Plan dual"
d_9^*
d_6^*
d_5^*
d_8^*
d_7^*
d_1^*
d_2^*
d_4^*
d_3^*
Configurația duală:
determinarea frontierei superioare a acoperirii convexe ptr. dualele ~~punctelor~~ dreptelor din LE
determinarea frontierei inferioare ptr. dualele dreptelor din UE
Duale LE:
$d_1^* = (1, -4)$
$d_2^* = (-1, -5)$
$d_3^* = (0, -8)$
$d_4^* = (\frac{1}{5}, -4.5)$
pentru aceste puncte, partea superioară a frontierei acop. convexe e determinată de d_1^* și d_2^*
Duale UE:
$d_5^* = (-3, 6)$
$d_6^* = (0, 7)$
$d_7^* = (0, 4)$
$d_8^* = (1, 5)$
$d_9^* = (-1, 8)$
ptr. aceste puncte, partea inferioară a fr. acop. convexe e dată de d_5^*, d_7^*, d_8^*

(ii) Abordarea calitativă. Motivație

- Sunt realizate 3 produse (notate 1, 2 și 3) pe 2 aparate (notate X și Y).

(ii) Abordarea calitativă. Motivație

- Sunt realizate 3 produse (notate 1, 2 și 3) pe 2 aparate (notate X și Y).
- Ciclul de producție este săptămânal (40h de lucru). Timpul de producție (în minute) pentru produs este indicat în tabel.

	X	Y	Obs.	Nr. prod.	Spațiu	Profit
1	10	27	pe ambele	x_1	0.1m^2	10
2	12	19	în paralel, simultan	x_2, respectiv y_2	0.2m^2	13
3	8	24	în paralel, simultan	x_3, respectiv y_3	0.05m^2	9

(ii) Abordarea calitativă. Motivație

- Sunt realizate 3 produse (notate 1, 2 și 3) pe 2 aparate (notate X și Y).
- Ciclul de producție este săptămânal (40h de lucru). Timpul de producție (în minute) pentru produs este indicat în tabel.

	X	Y	Obs.	Nr. prod.	Spațiu	Profit
1	10	27	pe ambele	x_1	0.1m^2	10
2	12	19	în paralel, simultan	x_2, respectiv y_2	0.2m^2	13
3	8	24	în paralel, simultan	x_3, respectiv y_3	0.05m^2	9

- Aparatele X și Y au un interval de mentenanță de 5%, respectiv 7% din timpul de lucru. Spațiul total de depozitare este de 50m^2.

(ii) Abordarea calitativă. Motivație

- Sunt realizate 3 produse (notate 1, 2 și 3) pe 2 aparate (notate X și Y).
- Ciclul de producție este săptămânal (40h de lucru). Timpul de producție (în minute) pentru produs este indicat în tabel.

	X	Y	Obs.	Nr. prod.	Spațiu	Profit
1	10	27	pe ambele	x_1	0.1m^2	10
2	12	19	în paralel, simultan	x_2, respectiv y_2	0.2m^2	13
3	8	24	în paralel, simultan	x_3, respectiv y_3	0.05m^2	9

- Aparatele X și Y au un interval de mentenanță de 5%, respectiv 7% din timpul de lucru. Spațiul total de depozitare este de 50m^2.
- Modelul matematic:

Constrângeri:

$$\begin{array}{ll} 0.1x_1 + 0.2(x_2 + y_2) + 0.05(x_3 + y_3) \leq 50 & \textit{Spațiu de depozitare} \\ 10x_1 + 12x_2 + 8x_3 \leq 0.95 \cdot 40 \cdot 60 & \textit{Timp aparatul } X \\ 27x_1 + 19y_2 + 24y_3 \leq 0.93 \cdot 40 \cdot 60 & \textit{Timp aparatul } Y \end{array}$$

Cerința:

$$\text{maximizează}(10x_1 + 13(x_2 + y_2) + 9(x_3 + y_3))$$

Problematizare, terminologie

- **Formulare generală (în spațiul *d*-dimensional):**

$$\text{maximizează}(c_1x_1 + c_2x_2 + \ldots + c_dx_d)$$

date constrângerile liniare (inegalități)

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \ldots a_{1d}x_d \leq b_1 \\ a_{21}x_1 + a_{22}x_2 + \ldots a_{2d}x_d \leq b_2 \\ \ldots \\ a_{n1}x_1 + a_{n2}x_2 + \ldots a_{nd}x_d \leq b_n \end{cases} \tag{1}$$

Problematizare, terminologie

- **Formulare generală (în spațiul *d*-dimensional):**

$$\text{maximizează}(c_1x_1 + c_2x_2 + \ldots + c_dx_d)$$

date constrângerile liniare (inegalități)

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \ldots a_{1d}x_d \leq b_1 \\ a_{21}x_1 + a_{22}x_2 + \ldots a_{2d}x_d \leq b_2 \\ \ldots \\ a_{n1}x_1 + a_{n2}x_2 + \ldots a_{nd}x_d \leq b_n \end{cases} \tag{1}$$

- **Denumiri:**

Problematizare, terminologie

- **Formulare generală (în spațiul d-dimensional):**

$$\text{maximizează}(c_1x_1 + c_2x_2 + \ldots + c_dx_d)$$

date constrângerile liniare (inegalități)

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \ldots a_{1d}x_d \leq b_1 \\ a_{21}x_1 + a_{22}x_2 + \ldots a_{2d}x_d \leq b_2 \\ \ldots \\ a_{n1}x_1 + a_{n2}x_2 + \ldots a_{nd}x_d \leq b_n \end{cases} \tag{1}$$

- **Denumiri:**
  - date de intrare: $(a_{ij})_{i=\overline{1,n},\, j=\overline{1,d}}, (b_i)_{i=\overline{1,n}}, (c_j)_{j=\overline{1,d}}$

Problematizare, terminologie

- **Formulare generală (în spațiul d-dimensional):**

$$\text{maximizează}(c_1x_1 + c_2x_2 + \ldots + c_dx_d)$$

date constrângerile liniare (inegalități)

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \ldots a_{1d}x_d \leq b_1 \\ a_{21}x_1 + a_{22}x_2 + \ldots a_{2d}x_d \leq b_2 \\ \ldots \\ a_{n1}x_1 + a_{n2}x_2 + \ldots a_{nd}x_d \leq b_n \end{cases} \quad (1)$$

- **Denumiri:**
  - date de intrare: $(a_{ij})_{i=\overline{1,n},\, j=\overline{1,d}}, (b_i)_{i=\overline{1,n}}, (c_j)_{j=\overline{1,d}}$
  - funcție obiectiv: $(c_1x_1 + c_2x_2 + \ldots + c_dx_d)$

Problematizare, terminologie

- **Formulare generală (în spațiul d-dimensional):**

$$\text{maximizează}(c_1x_1 + c_2x_2 + \ldots + c_dx_d)$$

date constrângerile liniare (inegalități)

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \ldots a_{1d}x_d \leq b_1 \\ a_{21}x_1 + a_{22}x_2 + \ldots a_{2d}x_d \leq b_2 \\ \ldots \\ a_{n1}x_1 + a_{n2}x_2 + \ldots a_{nd}x_d \leq b_n \end{cases} \tag{1}$$

- **Denumiri:**
  - date de intrare: $(a_{ij})_{i=\overline{1,n},\, j=\overline{1,d}}, (b_i)_{i=\overline{1,n}}, (c_j)_{j=\overline{1,d}}$
  - funcție obiectiv: $(c_1x_1 + c_2x_2 + \ldots + c_dx_d)$
  - constrângeri: inegalitățile (1)

Problematizare, terminologie

- **Formulare generală (în spațiul d-dimensional):**

$$\text{maximizează}(c_1x_1 + c_2x_2 + \ldots + c_dx_d)$$

date constrângerile liniare (inegalități)

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \ldots a_{1d}x_d \leq b_1 \\ a_{21}x_1 + a_{22}x_2 + \ldots a_{2d}x_d \leq b_2 \\ \ldots \\ a_{n1}x_1 + a_{n2}x_2 + \ldots a_{nd}x_d \leq b_n \end{cases} \tag{1}$$

- **Denumiri:**
  - date de intrare: $(a_{ij})_{i=\overline{1,n},\, j=\overline{1,d}}, (b_i)_{i=\overline{1,n}}, (c_j)_{j=\overline{1,d}}$
  - funcție obiectiv: $(c_1x_1 + c_2x_2 + \ldots + c_dx_d)$
  - constrângeri: inegalitățile (1)
  - regiune realizabilă (fezabilă): intersecția semispațiilor care definesc constrângerile problemei

Problematizare, terminologie

- **Formulare generală (în spațiul d-dimensional):**

$$\text{maximizează}(c_1x_1 + c_2x_2 + \ldots + c_dx_d)$$

date constrângerile liniare (inegalități)

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \ldots a_{1d}x_d \le b_1 \\ a_{21}x_1 + a_{22}x_2 + \ldots a_{2d}x_d \le b_2 \\ \ldots \\ a_{n1}x_1 + a_{n2}x_2 + \ldots a_{nd}x_d \le b_n \end{cases} \tag{1}$$

- **Denumiri:**
  - date de intrare: $(a_{ij})_{i=\overline{1,n},\, j=\overline{1,d}}, (b_i)_{i=\overline{1,n}}, (c_j)_{j=\overline{1,d}}$
  - funcție obiectiv: $(c_1x_1 + c_2x_2 + \ldots + c_dx_d)$
  - constrângeri: inegalitățile (1)
  - regiune realizabilă (fezabilă): intersecția semispațiilor care definesc constrângerile problemei
- **Obs. Interpretare a cerinței de maximizare:** Maximizarea funcției obiectiv revine la a determina un punct al cărui vector de poziție are proiecția maximă de direcția dată de vectorul $\vec{c} = (c_1, c_2, \ldots, c_d)$.

Exemplu - cazul 1D ($d = 1$)

funcție obiectiv

maximizează (cx)

Coordonata: x :

$$\begin{cases} a_1 x \le b_1 & \text{- interval} \\ a_2 x \le b_2 & \text{—"—} \\ \vdots & \\ a_n x \le b_n & \text{—"—} \end{cases}$$

constrângeri

<u>Exemplu concret</u>: maximizează $(2x)$

$$\begin{cases} 3x \le 6 \\ -2x \le 4 \\ 6x \le 6 \end{cases} \quad \begin{aligned} & x \le 2, \quad x \in (-\infty, 2] \\ & x \ge -2, \quad x \in [-2, \infty) \\ & x \le 1, \quad x \in (-\infty, 1] \end{aligned}$$

intervalul $[-2, 1]$ regiune fezabilă

Maximul funcției obiectiv este egal cu 2 și se obține ptr $\underline{x = 1}$.

Exemplu - cazul 1D ($d = 1$)

funcție obiectiv

Coordonata: x : maximizează (cx)

$$\begin{cases} a_1 x \le b_1 & \text{- interval} \\ a_2 x \le b_2 & \text{---} \\ \vdots & \\ a_n x \le b_n & \text{---} \end{cases}$$

$\to$ constrângeri

Exemplu concret: maximizează $(2x)$

$$\begin{cases} 3x \le 6 \\ -2x \le 4 \\ 6x \le 6 \end{cases} \quad \begin{array}{ll} x \le 2, & x \in (-\infty, 2] \\ x \ge -2, & x \in [-2, \infty) \\ x \le 1, & x \in (-\infty, 1] \end{array}$$

intervalul $[-2, 1]$ regiune fezabilă

Maximul funcției obiectiv este egal cu 2 și se obține ptr $x = 1$.

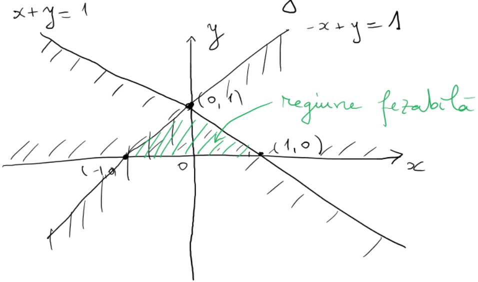
Lemă. (Pentru $d = 1$) *Un program liniar 1-dimensional poate fi rezolvat în timp liniar.*

Exemplu - cazul 2D ($d = 2$)

Notăm coordonatele cu x și y.

maximizează (y) ; $\vec{c} = (0, 1)$, date $\begin{cases} x + y \leq 1 \\ -y \leq 0 \\ -x + y \leq 1 \end{cases}$

$y = 0 \cdot x + 1 \cdot y$



Funcția are valoarea maximă 1, atinsă în punctul $(0, 1)$.

Probleme de programare liniară în plan ($d = 2$)

- **Convenții și terminologie:**

Probleme de programare liniară în plan ($d = 2$)

- **Convenții și terminologie:**
  - Coordonatele: x și y

Probleme de programare liniară în plan ($d = 2$)

- **Convenții și terminologie:**
  - Coordonatele: x și y
  - Funcția obiectiv: $f_{\vec{c}}(p) = c_x x + c_y y$, unde $\vec{c} = (c_x, c_y)$.

Probleme de programare liniară în plan ($d = 2$)

- **Convenții și terminologie:**
  - Coordonatele: x și y
  - Funcția obiectiv: $f_{\overrightarrow{c}}(p) = c_x x + c_y y$, unde $\overrightarrow{c} = (c_x, c_y)$.
  - Constrângerile: $h_1, h_2, \ldots, h_n$ (semiplane); se notează $H = \{h_1, h_2, \ldots, h_n\}$
  - Regiunea fezabilă este $C = h_1 \cap h_2 \cap \ldots \cap h_n$.

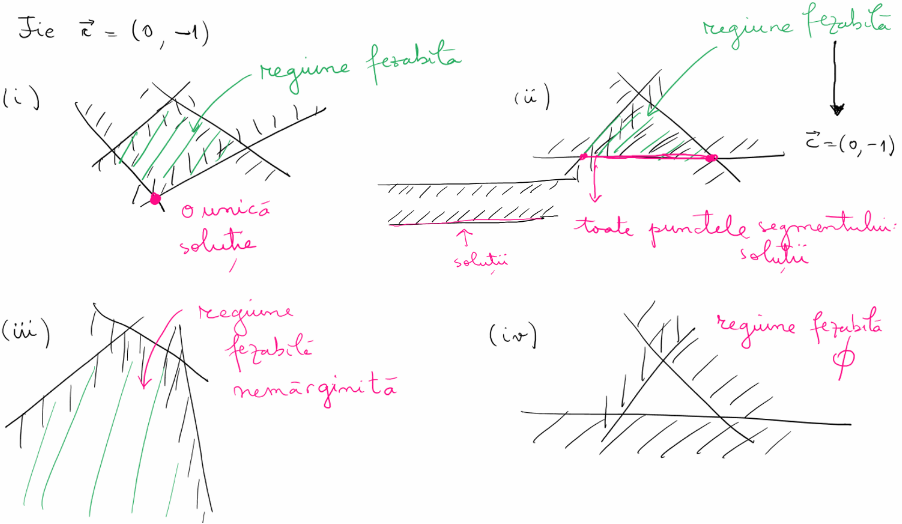
Probleme de programare liniară în plan ($d = 2$)

- **Convenții și terminologie:**
  - Coordonatele: x și y
  - Funcția obiectiv: $f_{\vec{c}}(p) = c_x x + c_y y$, unde $\vec{c} = (c_x, c_y)$.
  - Constrângerile: $h_1, h_2, \ldots, h_n$ (semiplane); se notează $H = \{h_1, h_2, \ldots, h_n\}$
  - Regiunea fezabilă este $C = h_1 \cap h_2 \cap \ldots \cap h_n$.
- **Program liniar:** $(H, \vec{c})$.
- **Scop:** Se caută $p \in C$ astfel ca $f_{\vec{c}}(p)$ să fie maximă.

Probleme de programare liniară în plan ($d = 2$)

- **Convenții și terminologie:**
  - Coordonatele: x și y
  - Funcția obiectiv: $f_{\vec{c}}(p) = c_x x + c_y y$, unde $\vec{c} = (c_x, c_y)$.
  - Constrângerile: $h_1, h_2, \ldots, h_n$ (semiplane); se notează $H = \{h_1, h_2, \ldots, h_n\}$
  - Regiunea fezabilă este $C = h_1 \cap h_2 \cap \ldots \cap h_n$.
  - **Program liniar:** $(H, \vec{c})$.
  - **Scop:** Se caută $p \in C$ astfel ca $f_{\vec{c}}(p)$ să fie maximă.
- Pentru o problemă de programare liniară în plan pot fi distinse patru situații: (i) o soluție unică; (ii) toate punctele de pe o muchie sunt soluții; (iii) regiunea fezabilă este nemărginită și pot fi găsite soluții de-a lungul unei semidrepte; (iv) regiunea fezabilă este vidă.

Cazul 2D ($d = 2$) - exemple de regiuni fezabile



Algoritm incremental pentru rezolvarea unei probleme de programare liniară 2D

- Principii:

Algoritm incremental pentru rezolvarea unei probleme de programare liniară 2D

- Principii:
  - constrângerile sunt adăugate una câte una;

Algoritm incremental pentru rezolvarea unei probleme de programare liniară 2D

- Principii:
  - constrângerile sunt adăugate una câte una;
  - presupunem că la fiecare pas soluția (punctul de maxim) există, apoi actualizează;
  - sunt adăugate la început constrângeri care garantează mărginirea programului liniar, definite astfel: se alege $M >> 0$ și se definesc noi constrângeri convenabile;

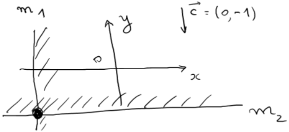
Algoritm incremental pentru rezolvarea unei probleme de programare liniară 2D

- Principii:
  - constrângerile sunt adăugate una câte una;
  - presupunem că la fiecare pas soluția (punctul de maxim) există, apoi actualizează;
  - sunt adăugate la început constrângeri care garantează mărginirea programului liniar, definite astfel: se alege $M >> 0$ și se definesc noi constrângeri convenabile;
  - se lucrează cu convenția de ordonare lexicografică, astfel încât există o **unică** soluție optimă.

Algoritm incremental pentru rezolvarea unei probleme de programare liniară 2D

- Principii:
  - constrângerile sunt adăugate una câte una;
  - presupunem că la fiecare pas soluția (punctul de maxim) există, apoi actualizează;
  - sunt adăugate la început constrângeri care garantează mărginirea programului liniar, definite astfel: se alege $M >> 0$ și se definesc noi constrângeri convenabile;
  - se lucrează cu convenția de ordonare lexicografică, astfel încât există o **unică** soluție optimă.
- Vom considera în continuare $\vec{c} = (0, -1)$, iar noile constrângeri vor fi:

$$m_1 : x \geq -M, \qquad m_2 : y \geq -M.$$



Notații

- Fie $(H, \overrightarrow{c})$ un program liniar cu constrângerile $h_1, h_2, \ldots, h_n$. Se notează:

$$H_i = \{m_1, m_2, h_1, h_2, \ldots, h_i\}, \text{ mulțime de semiplane}$$

$$C_i = m_1 \cap m_2 \cap h_1 \cap h_2 \cap \ldots \cap h_i, \text{ regiune fezabilă.}$$

Notația este pentru $i = 0, \ldots, n$, în particular

$$H_0 = \{m_1, m_2\} \qquad C_0 = m_1 \cap m_2.$$

Notații

- Fie $(H, \overrightarrow{c})$ un program liniar cu constrângerile $h_1, h_2, \ldots, h_n$. Se notează:

$$H_i = \{m_1, m_2, h_1, h_2, \ldots, h_i\}, \text{ mulțime de semiplane}$$

$$C_i = m_1 \cap m_2 \cap h_1 \cap h_2 \cap \ldots \cap h_i, \text{ regiune fezabilă.}$$

Notația este pentru $i = 0, \ldots, n$, în particular

$$H_0 = \{m_1, m_2\} \qquad C_0 = m_1 \cap m_2.$$

- Observații:

Notații

- Fie $(H, \overrightarrow{c})$ un program liniar cu constrângerile $h_1, h_2, \ldots, h_n$. Se notează:

$$H_i = \{m_1, m_2, h_1, h_2, \ldots, h_i\}, \text{ mulțime de semiplane}$$

$$C_i = m_1 \cap m_2 \cap h_1 \cap h_2 \cap \ldots \cap h_i, \text{ regiune fezabilă.}$$

Notația este pentru $i = 0, \ldots, n$, în particular

$$H_0 = \{m_1, m_2\} \qquad C_0 = m_1 \cap m_2.$$

- Observații:
  (i) $C_0 \supseteq C_1 \supseteq C_2 \supseteq \ldots \supseteq C_n = C$.

Notații

- Fie $(H, \vec{c})$ un program liniar cu constrângerile $h_1, h_2, \ldots, h_n$. Se notează:

$$H_i = \{m_1, m_2, h_1, h_2, \ldots, h_i\}, \text{ mulțime de semiplane}$$

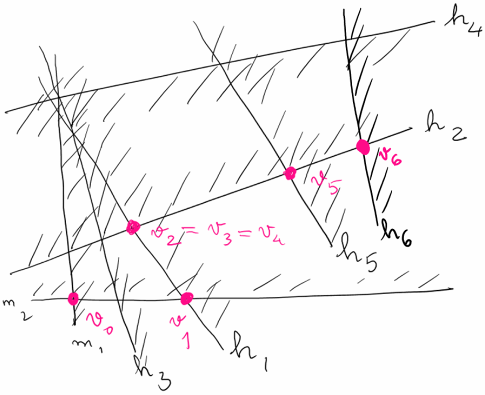
$$C_i = m_1 \cap m_2 \cap h_1 \cap h_2 \cap \ldots \cap h_i, \text{ regiune fezabilă.}$$

Notația este pentru $i = 0, \ldots, n$, în particular

$$H_0 = \{m_1, m_2\} \qquad C_0 = m_1 \cap m_2.$$

- Observații:
  (i) $C_0 \supseteq C_1 \supseteq C_2 \supseteq \ldots \supseteq C_n = C$.
  (ii) Pentru fiecare i, regiunea fezabilă C_i, dacă este nevidă, are un vârf care reprezintă o soluție optimă a problemei $(H_i, \vec{c})$. Punctul este notat cu v_i (depinde de alegerea lui m_1 și m_2).

Exemplu



$\vec{c} = (0,-1)$

$v_3 = v_2$, pt. că $v_2 \in h_3$

$v_4 = v_3 = v_2$ pt. că $v_3 \in h_4$

$v_5 \neq v_4$ pt. că $v_4 \notin h_5$

Observații

- Fie $1 \leq i \leq n$, presupunem că C_{i-1} și v_{i-1} sunt determinate. Considerăm h_i. Sunt două situații:

Observații

- Fie $1 \leq i \leq n$, presupunem că C_{i-1} și v_{i-1} sunt determinate. Considerăm h_i. Sunt două situații:
  (i) dacă $v_{i-1} \in h_i$, atunci $v_i = v_{i-1}$,

Observații

- Fie $1 \leq i \leq n$, presupunem că C_{i-1} și v_{i-1} sunt determinate. Considerăm h_i. Sunt două situații:
  (i) dacă $v_{i-1} \in h_i$, atunci $v_i = v_{i-1}$,
  (ii) dacă $v_{i-1} \notin h_i$, atunci

Observații

- Fie $1 \leq i \leq n$, presupunem că C_{i-1} și v_{i-1} sunt determinate. Considerăm h_i. Sunt două situații:
  (i) dacă $v_{i-1} \in h_i$, atunci $v_i = v_{i-1}$,
  (ii) dacă $v_{i-1} \notin h_i$, atunci

    fie $C_i = \emptyset$

Observații

- Fie $1 \leq i \leq n$, presupunem că C_{i-1} și v_{i-1} sunt determinate. Considerăm h_i. Sunt două situații:
  (i) dacă $v_{i-1} \in h_i$, atunci $v_i = v_{i-1}$,
  (ii) dacă $v_{i-1} \notin h_i$, atunci

  fie $C_i = \emptyset$

  fie $v_i \in d_i$, unde d_i este dreapta care mărginește h_i.

Observații

- Fie $1 \leq i \leq n$, presupunem că C_{i-1} și v_{i-1} sunt determinate. Considerăm h_i. Sunt două situații:
  (i) dacă $v_{i-1} \in h_i$, atunci $v_i = v_{i-1}$,
  (ii) dacă $v_{i-1} \notin h_i$, atunci

  fie $C_i = \emptyset$

  fie $v_i \in d_i$, unde d_i este dreapta care mărginește h_i.

  În acest caz, găsirea lui v_i revine la găsirea lui $p \in d_i$ care maximizează $f_{\vec{c}}(p)$, date constrângerile deja existente ($p \in h, \forall h \in H_i$). **De fapt, aceasta este o problemă pe programare liniară 1-dimensională, care are complexitatea-timp liniară, adică $O(i)$.**

Algoritm LPMARG2D $(H, \overrightarrow{c}, m_1, m_2)$

- **Input.** Un program liniar $(H \cup \{m_1, m_2\}, \overrightarrow{c})$ din $\mathbb{R}^2$

Algoritm LPMARG2D $(H, \overrightarrow{c}, m_1, m_2)$

- **Input.** Un program liniar $(H \cup \{m_1, m_2\}, \overrightarrow{c})$ din $\mathbb{R}^2$
- **Output.** Dacă $(H \cup \{m_1, m_2\}, \overrightarrow{c})$ nu e realizabil (fezabil), raportează. În caz contrar, indică punctul cel mai mic lexicografic p care maximizează $f_{\overrightarrow{c}}(p)$.

Algoritm LPMARG2D $(H, \overrightarrow{c}, m_1, m_2)$

- **Input.** Un program liniar $(H \cup \{m_1, m_2\}, \overrightarrow{c})$ din $\mathbb{R}^2$
- **Output.** Dacă $(H \cup \{m_1, m_2\}, \overrightarrow{c})$ nu e realizabil (fezabil), raportează. În caz contrar, indică punctul cel mai mic lexicografic p care maximizează $f_{\overrightarrow{c}}(p)$.

1. $v_0 \leftarrow$ "colțul" lui c_0

Algoritm LPMARG2D $(H, \overrightarrow{c}, m_1, m_2)$

- **Input.** Un program liniar $(H \cup \{m_1, m_2\}, \overrightarrow{c})$ din $\mathbb{R}^2$
- **Output.** Dacă $(H \cup \{m_1, m_2\}, \overrightarrow{c})$ nu e realizabil (fezabil), raportează. În caz contrar, indică punctul cel mai mic lexicografic p care maximizează $f_{\overrightarrow{c}}(p)$.

1. $v_0 \leftarrow$ "colțul" lui c_0
2. fie $h_1, h_2, \ldots, h_n$ semiplanele din H

Algoritm LPMARG2D $(H, \vec{c}, m_1, m_2)$

- **Input.** Un program liniar $(H \cup \{m_1, m_2\}, \vec{c})$ din $\mathbb{R}^2$
- **Output.** Dacă $(H \cup \{m_1, m_2\}, \vec{c})$ nu e realizabil (fezabil), raportează. În caz contrar, indică punctul cel mai mic lexicografic p care maximizează $f_{\vec{c}}(p)$.

1. $v_0 \leftarrow$ "colțul" lui c_0
2. fie $h_1, h_2, \ldots, h_n$ semiplanele din H
3. **for** $i \leftarrow 1$ **to** n

Algoritm LPMARG2D $(H, \vec{c}, m_1, m_2)$

- **Input.** Un program liniar $(H \cup \{m_1, m_2\}, \vec{c})$ din $\mathbb{R}^2$
- **Output.** Dacă $(H \cup \{m_1, m_2\}, \vec{c})$ nu e realizabil (fezabil), raportează. În caz contrar, indică punctul cel mai mic lexicografic p care maximizează $f_{\vec{c}}(p)$.

1. $v_0 \leftarrow$ "colțul" lui c_0
2. fie $h_1, h_2, \ldots, h_n$ semiplanele din H
3. **for** $i \leftarrow 1$ **to** n
4.   **do if** $v_{i-1} \in h_i$

Algoritm LPMARG2D $(H, \vec{c}, m_1, m_2)$

- **Input.** Un program liniar $(H \cup \{m_1, m_2\}, \vec{c})$ din $\mathbb{R}^2$
- **Output.** Dacă $(H \cup \{m_1, m_2\}, \vec{c})$ nu e realizabil (fezabil), raportează. În caz contrar, indică punctul cel mai mic lexicografic p care maximizează $f_{\vec{c}}(p)$.

1. $v_0 \leftarrow$ "colțul" lui c_0
2. fie $h_1, h_2, \ldots, h_n$ semiplanele din H
3. **for** $i \leftarrow 1$ **to** n
4.   **do if** $v_{i-1} \in h_i$
5.   **then** $v_i \leftarrow v_{i-1}$

Algoritm LPMARG2D $(H, \vec{c}, m_1, m_2)$

- **Input.** Un program liniar $(H \cup \{m_1, m_2\}, \vec{c})$ din $\mathbb{R}^2$
- **Output.** Dacă $(H \cup \{m_1, m_2\}, \vec{c})$ nu e realizabil (fezabil), raportează. În caz contrar, indică punctul cel mai mic lexicografic p care maximizează $f_{\vec{c}}(p)$.

1. $v_0 \leftarrow$ "colțul" lui c_0
2. fie $h_1, h_2, \ldots, h_n$ semiplanele din H
3. **for** $i \leftarrow 1$ **to** n
4. **do if** $v_{i-1} \in h_i$
5. **then** $v_i \leftarrow v_{i-1}$
6. **else** $v_i \leftarrow$ punctul p de pe d_i care maximizează $f_{\vec{c}}(p)$ date constrângerile din H_i

Algoritm LPMARG2D $(H, \vec{c}, m_1, m_2)$

- **Input.** Un program liniar $(H \cup \{m_1, m_2\}, \vec{c})$ din $\mathbb{R}^2$
- **Output.** Dacă $(H \cup \{m_1, m_2\}, \vec{c})$ nu e realizabil (fezabil), raportează. În caz contrar, indică punctul cel mai mic lexicografic p care maximizează $f_{\vec{c}}(p)$.

1. $v_0 \leftarrow$ "colțul" lui c_0
2. fie $h_1, h_2, \ldots, h_n$ semiplanele din H
3. **for** $i \leftarrow 1$ **to** n
4. **do if** $v_{i-1} \in h_i$
5. **then** $v_i \leftarrow v_{i-1}$
6. **else** $v_i \leftarrow$ punctul p de pe d_i care maximizează $f_{\vec{c}}(p)$ date constrângerile din H_i
7. **if** p nu există

Algoritm LPMARG2D $(H, \vec{c}, m_1, m_2)$

- **Input.** Un program liniar $(H \cup \{m_1, m_2\}, \vec{c})$ din $\mathbb{R}^2$
- **Output.** Dacă $(H \cup \{m_1, m_2\}, \vec{c})$ nu e realizabil (fezabil), raportează. În caz contrar, indică punctul cel mai mic lexicografic p care maximizează $f_{\vec{c}}(p)$.

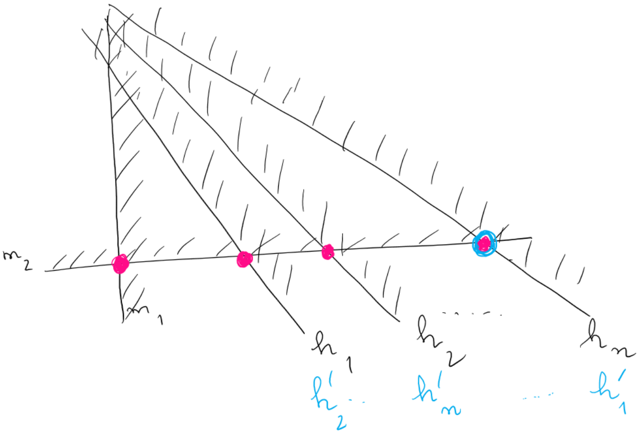
1. $v_0 \leftarrow$ "colțul" lui c_0
2. fie $h_1, h_2, \ldots, h_n$ semiplanele din H
3. **for** $i \leftarrow 1$ **to** n
4.   **do if** $v_{i-1} \in h_i$
5.   **then** $v_i \leftarrow v_{i-1}$
6.   **else** $v_i \leftarrow$ punctul p de pe d_i care maximizează $f_{\vec{c}}(p)$ date constrângerile din H_i
7.   **if** p nu există
8.   **then** raportează "nefezabil" **end**

Algoritm LPMARG2D $(H, \vec{c}, m_1, m_2)$

- **Input.** Un program liniar $(H \cup \{m_1, m_2\}, \vec{c})$ din $\mathbb{R}^2$
- **Output.** Dacă $(H \cup \{m_1, m_2\}, \vec{c})$ nu e realizabil (fezabil), raportează. În caz contrar, indică punctul cel mai mic lexicografic p care maximizează $f_{\vec{c}}(p)$.

1. $v_0 \leftarrow$ "colțul" lui c_0
2. fie $h_1, h_2, \ldots, h_n$ semiplanele din H
3. **for** $i \leftarrow 1$ **to** n
4.   **do if** $v_{i-1} \in h_i$
5.   **then** $v_i \leftarrow v_{i-1}$
6.   **else** $v_i \leftarrow$ punctul p de pe d_i care maximizează $f_{\vec{c}}(p)$ date constrângerile din H_i
7.   **if** p nu există
8.   **then** raportează "nefezabil" **end**
9. **return** v_n

Comentariu - ordinea contează



Algoritm aleatoriu

- Pasul **2.** este înlocuit cu:
  2'. Calculează o permutare arbitrară a semiplanelor, folosind o procedură adecvată.

Algoritm aleatoriu

- Pasul **2.** este înlocuit cu:
  2'. Calculează o permutare arbitrară a semiplanelor, folosind o procedură adecvată.
- Algoritmul incremental $\mathrm{LPMARG2D}$ are complexitate-timp $O(n^2)$, iar varianta bazată pe alegerea aleatorie a semiplanelor are complexitate-timp medie $O(n)$ (n este numărul semiplanelor).

Analiza complexității-timp - varianta algoritmului probabilist (I)

Fie $(X_i)_{i=1,\ldots,n}$ variabilă aleatoare definită astfel:

$$X_i = \begin{cases} 0, & \text{dacă } v_{i-1} \in h_i \text{ (adică este ales pasul 5 la iterația } i\text{)} \\ 1, & \text{dacă } v_{i-1} \notin h_i \text{ (adică este ales pasul 6 la iterația } i\text{).} \end{cases}$$

Analiza complexității-timp - varianta algoritmului probabilist (I)

Fie $(X_i)_{i=1,\ldots,n}$ variabilă aleatoare definită astfel:

$$X_i = \begin{cases} 0, & \text{dacă } v_{i-1} \in h_i \text{ (adică este ales pasul 5 la iterația } i\text{)} \\ 1, & \text{dacă } v_{i-1} \notin h_i \text{ (adică este ales pasul 6 la iterația } i\text{)}. \end{cases}$$

În concluzie, timpul total de rulare este $\sum_{i=1}^{n} X_i O(i) + O(n)$.

Analiza complexității-timp - varianta algoritmului probabilist (I)

Fie $(X_i)_{i=1,\dots,n}$ variabilă aleatoare definită astfel:

$$X_i = \begin{cases} 0, & \text{dacă } v_{i-1} \in h_i \text{ (adică este ales pasul 5 la iterația } i) \\ 1, & \text{dacă } v_{i-1} \notin h_i \text{ (adică este ales pasul 6 la iterația } i). \end{cases}$$

În concluzie, timpul total de rulare este $\sum_{i=1}^{n} X_i O(i) + O(n)$.

Valoarea așteptată (timpul mediu)

$$E[\sum_{i=1}^{n} X_i O(i)] = \mu(\sum_{i=1}^{n} X_i O(i)) = \sum_{i=1}^{n} O(i)\mu(X_i) \leq \sum_{i=1}^{n} O(i) \cdot \frac{2}{i} = O(n).$$

Analiza complexității-timp - varianta algoritmului probabilist (I)

Fie $(X_i)_{i=1,\ldots,n}$ variabilă aleatoare definită astfel:

$$X_i = \begin{cases} 0, & \text{dacă } v_{i-1} \in h_i \text{ (adică este ales pasul 5 la iterația } i\text{)} \\ 1, & \text{dacă } v_{i-1} \notin h_i \text{ (adică este ales pasul 6 la iterația } i\text{).} \end{cases}$$

În concluzie, timpul total de rulare este $\sum_{i=1}^{n} X_i O(i) + O(n)$.

Valoarea așteptată (timpul mediu)

$$E[\sum_{i=1}^{n} X_i O(i)] = \mu(\sum_{i=1}^{n} X_i O(i)) = \sum_{i=1}^{n} O(i)\mu(X_i) \leq \sum_{i=1}^{n} O(i) \cdot \frac{2}{i} = O(n).$$

Vom arăta că $\mu(X_i) \leq \frac{2}{i}$, de unde va rezulta inegalitatea dorită.

Analiza complexității-timp - varianta algoritmului probabilist (II)

- Demonstrăm că $\mu(X_i) \leq \frac{2}{i}$, pentru orice $i = 1, \ldots, n$, adică probabilitatea ca $v_{i-1} \notin h_i$ este $\leq \frac{2}{i}$.

Analiza complexității-timp - varianta algoritmului probabilist (II)

- Demonstrăm că $\mu(X_i) \leq \frac{2}{i}$, pentru orice $i = 1, \ldots, n$, adică probabilitatea ca $v_{i-1} \notin h_i$ este $\leq \frac{2}{i}$.
- Arătăm inegalitatea pentru $i = n$ (cazul general, analog). Presupunem algoritmul terminat, v_n vârful optim.

Analiza complexității-timp - varianta algoritmului probabilist (II)

- Demonstrăm că $\mu(X_i) \leq \frac{2}{i}$, pentru orice $i = 1, \ldots, n$, adică probabilitatea ca $v_{i-1} \notin h_i$ este $\leq \frac{2}{i}$.
- Arătăm inegalitatea pentru $i = n$ (cazul general, analog). Presupunem algoritmul terminat, v_n vârful optim.
    - Care este probabilitatea ca $v_{n-1} \notin h_n$, adică la adăugarea lui h_n, vârful v_{n-1} să fie modificat în v_n? $\Leftrightarrow$

Analiza complexității-timp - varianta algoritmului probabilist (II)

- Demonstrăm că $\mu(X_i) \leq \frac{2}{i}$, pentru orice $i = 1, \ldots, n$, adică probabilitatea ca $v_{i-1} \notin h_i$ este $\leq \frac{2}{i}$.
- Arătăm inegalitatea pentru $i = n$ (cazul general, analog). Presupunem algoritmul terminat, v_n vârful optim.
  - Care este probabilitatea ca $v_{n-1} \notin h_n$, adică la adăugarea lui h_n, vârful v_{n-1} să fie modificat în v_n? $\Leftrightarrow$
  - Care este probabilitatea ca eliminând unul dintre semiplane să fie modificat vârful optim v_n?

Analiza complexității-timp - varianta algoritmului probabilist (II)

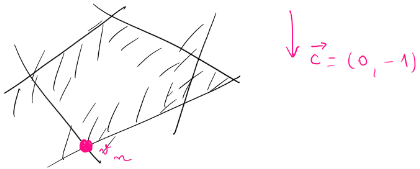
- Demonstrăm că $\mu(X_i) \leq \frac{2}{i}$, pentru orice $i = 1, \ldots, n$, adică probabilitatea ca $v_{i-1} \notin h_i$ este $\leq \frac{2}{i}$.
- Arătăm inegalitatea pentru $i = n$ (cazul general, analog). Presupunem algoritmul terminat, v_n vârful optim.
  - Care este probabilitatea ca $v_{n-1} \notin h_n$, adică la adăugarea lui h_n, vârful v_{n-1} să fie modificat în v_n? $\Leftrightarrow$
  - Care este probabilitatea ca eliminând unul dintre semiplane să fie modificat vârful optim v_n?

· nr. cazuri posibile: n

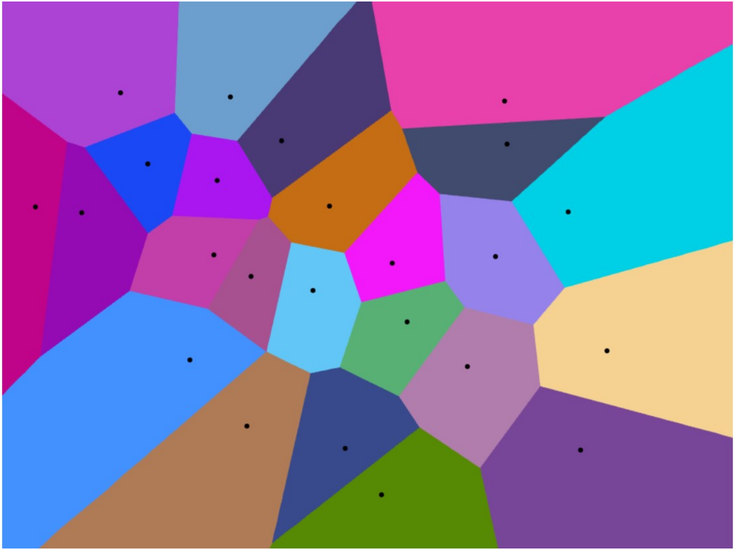
· nr. cazuri survine modificare: 2

$\Rightarrow$ probabilitatea de a modifica $\leq \frac{2}{n}$.

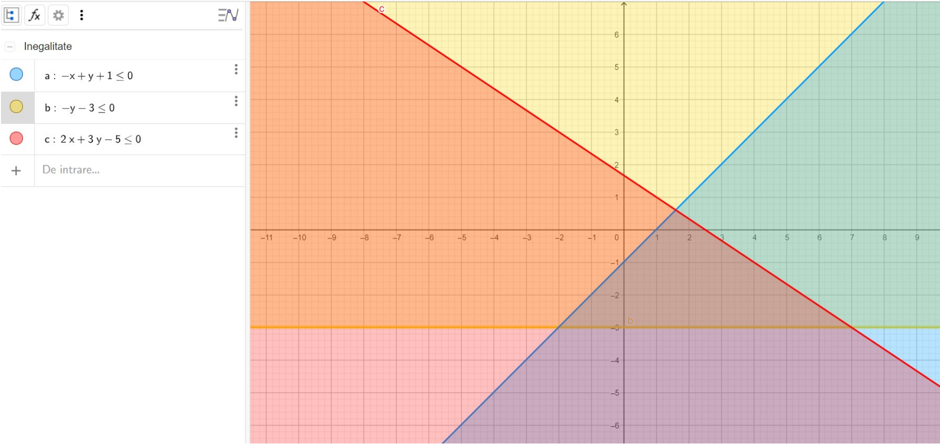
analog ptr. i. <u>q.e.d.</u>



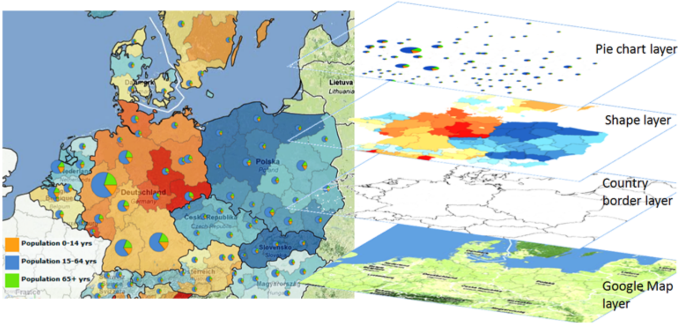
Motivație - reprezentarea diagramelor Voronoi



Motivație - reprezentarea intersecțiilor de semiplane



Motivație - reprezentarea datelor geo-spațiale

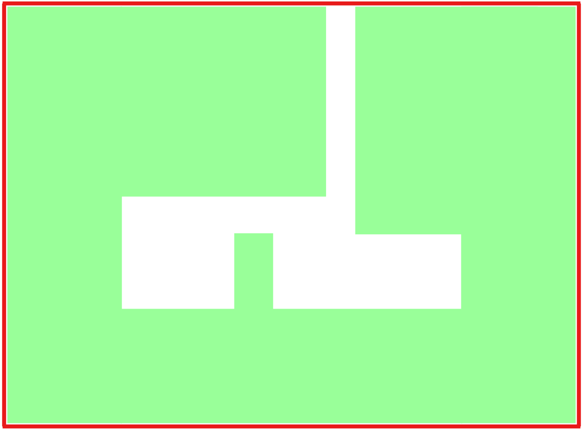


Sursa: https://s-media-cache-ak0.pinimg.com/originals/37/90/86/37908600ab7db99c424c3bc6e1ddb740.jpg

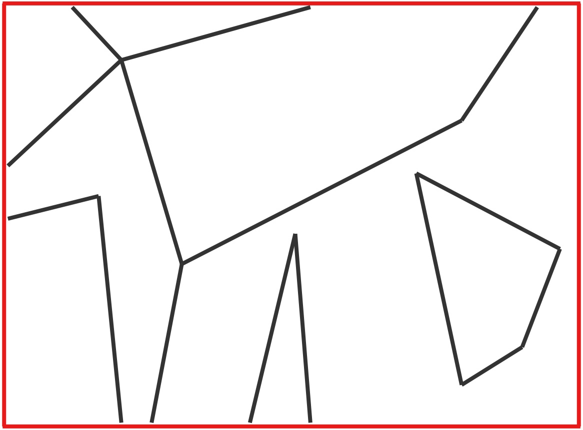
Ce structură de date este adecvată pentru a memora astfel de informații?

Problematizare

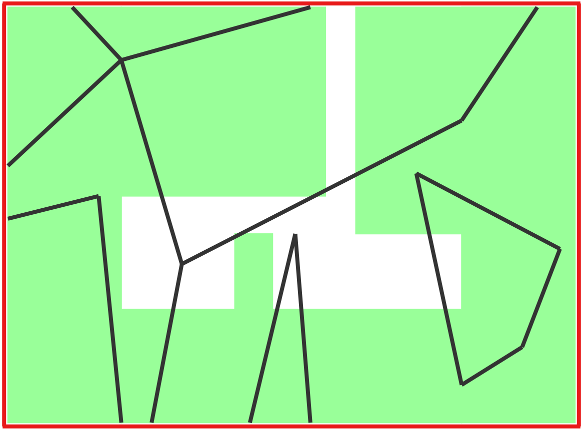
Problematizare



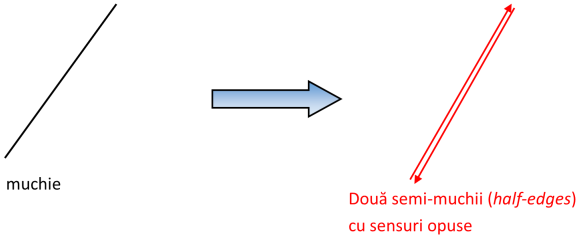
Problematizare



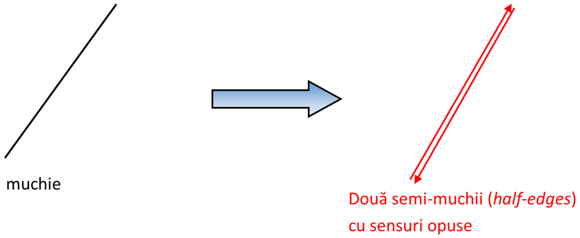
Problematizare



Ideea cheie

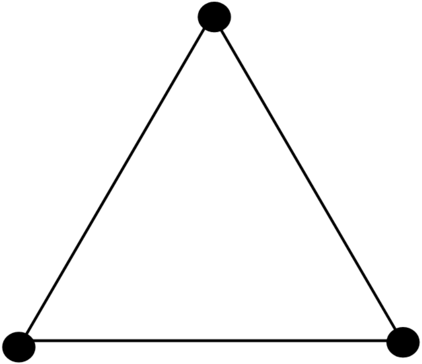


Ideea cheie

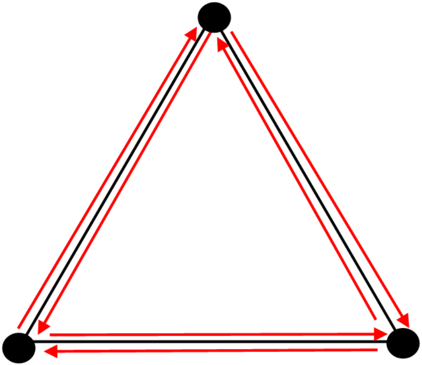


- Ideea fundamentală este de a introduce conceptul de semi-muchie (muchie orientată), cf. *"half-edge"*.

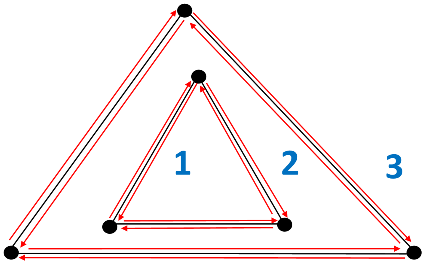
Exemplu



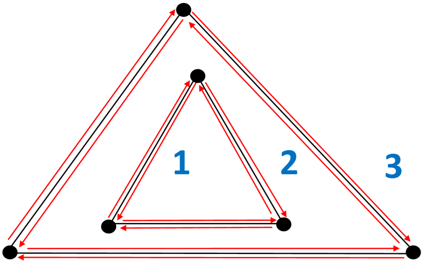
Exemplu



Exemplu

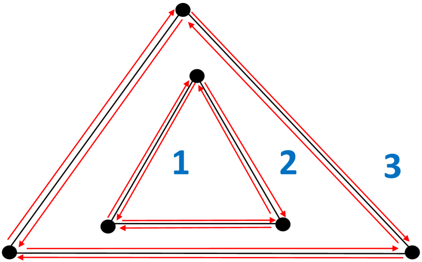


Exemplu



Subdiviziunea din figură are 6 vârfuri, 12 semi-muchii, 3 fețe.

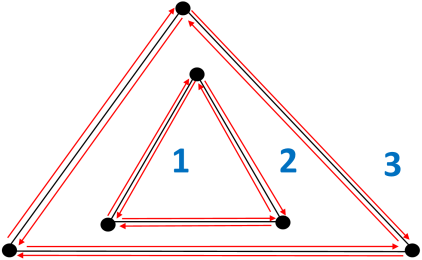
Exemplu



Subdiviziunea din figură are 6 vârfuri, 12 semi-muchii, 3 fețe.

Semi-muchiile delimitează frontiere ale fețelor, fața delimitată fiind la stânga.

Exemplu

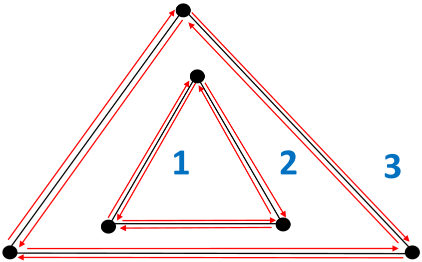


Subdiviziunea din figură are 6 vârfuri, 12 semi-muchii, 3 fețe.

Semi-muchiile delimitează frontiere ale fețelor, fața delimitată fiind la stânga.

Dat un poligon (eventual cu goluri):

Exemplu



Subdiviziunea din figură are 6 vârfuri, 12 semi-muchii, 3 fețe.

Semi-muchiile delimitează frontiere ale fețelor, fața delimitată fiind la stânga.

Dat un poligon (eventual cu goluri):

- **frontieră exterioară**, care poate fi parcursă cu ajutorul semi-muchiilor astfel încât poligonul să fie la stânga frontierei, iar virajele convexe să fie la stânga,
- **frontieră interioară** (dacă există goluri), caz în care poligonul este tot la stânga, dar virajele în vârfurile convexe sunt la dreapta.

Subdiviziuni planare

- Conceptul de subdiviziune planară: vârfuri, muchii, fețe.

Subdiviziuni planare

- Conceptul de subdiviziune planară: vârfuri, muchii, fețe.
- **Listă de muchii dublu înlănțuite / DCEL - Doubly Connected Edge List** [Müller și Preparata, 1978] (trei înregistrări: vârfuri, fețe, muchii orientate (semi-muchii)).

Subdiviziuni planare

- Conceptul de subdiviziune planară: vârfuri, muchii, fețe.
- **Listă de muchii dublu înlănțuite / DCEL - Doubly Connected Edge List** [Müller și Preparata, 1978] (trei înregistrări: vârfuri, fețe, muchii orientate (semi-muchii)).
  - **Vârf** v: coordonatele lui v în $Coordinates(v)$, pointer $IncidentEdge(v)$ spre o muchie orientată care are v ca origine

Subdiviziuni planare

- Conceptul de subdiviziune planară: vârfuri, muchii, fețe.
- **Listă de muchii dublu înlănțuite / DCEL - Doubly Connected Edge List** [Müller și Preparata, 1978] (trei înregistrări: vârfuri, fețe, muchii orientate (semi-muchii)).
  - **Vârf** v: coordonatele lui v în $Coordinates(v)$, pointer $IncidentEdge(v)$ spre o muchie orientată care are v ca origine
  - **Față** f: pointer $OuterComponent(f)$ spre o muchie orientată corespunzătoare frontierei externe (pentru fața nemărginită este **nil**); listă $InnerComponents(f)$, care conține, pentru fiecare gol, un pointer către una dintre muchiile orientate de pe frontieră

Subdiviziuni planare

- Conceptul de subdiviziune planară: vârfuri, muchii, fețe.
- **Listă de muchii dublu înlănțuite / DCEL - Doubly Connected Edge List** [Müller și Preparata, 1978] (trei înregistrări: vârfuri, fețe, muchii orientate (semi-muchii)).
  - **Vârf** v: coordonatele lui v în $Coordinates(v)$, pointer $IncidentEdge(v)$ spre o muchie orientată care are v ca origine
  - **Față** f: pointer $OuterComponent(f)$ spre o muchie orientată corespunzătoare frontierei externe (pentru fața nemărginită este **nil**); listă $InnerComponents(f)$, care conține, pentru fiecare gol, un pointer către una dintre muchiile orientate de pe frontieră
  - **Muchie orientată** $\overrightarrow{e}$: pointer $Origin(\overrightarrow{e})$, pointer $Twin(\overrightarrow{e})$ pointer $IncidentFace(\overrightarrow{e})$, pointer $Next(\overrightarrow{e})$, pointer $Prev(\overrightarrow{e})$.

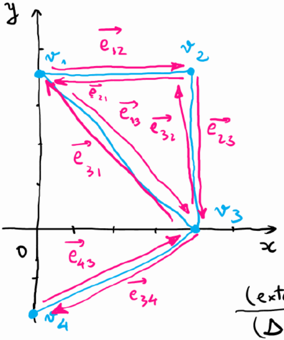
Subdiviziuni planare

- Conceptul de subdiviziune planară: vârfuri, muchii, fețe.
- **Listă de muchii dublu înlănțuite / DCEL - Doubly Connected Edge List** [Müller și Preparata, 1978] (trei înregistrări: vârfuri, fețe, muchii orientate (semi-muchii)).
  - **Vârf** v: coordonatele lui v în $Coordinates(v)$, pointer $IncidentEdge(v)$ spre o muchie orientată care are v ca origine
  - **Față** f: pointer $OuterComponent(f)$ spre o muchie orientată corespunzătoare frontierei externe (pentru fața nemărginită este **nil**); listă $InnerComponents(f)$, care conține, pentru fiecare gol, un pointer către una dintre muchiile orientate de pe frontieră
  - **Muchie orientată** $\overrightarrow{e}$: pointer $Origin(\overrightarrow{e})$, pointer $Twin(\overrightarrow{e})$ pointer $IncidentFace(\overrightarrow{e})$, pointer $Next(\overrightarrow{e})$, pointer $Prev(\overrightarrow{e})$.
- Oricărei subdiviziuni planare $\mathscr{S}$ i se asociază o listă de muchii dublu înlănțuite $\mathcal{D}_{\mathscr{S}}$.

Subdiviziuni planare

- Conceptul de subdiviziune planară: vârfuri, muchii, fețe.
- **Listă de muchii dublu înlănțuite / DCEL - Doubly Connected Edge List** [Müller și Preparata, 1978] (trei înregistrări: vârfuri, fețe, muchii orientate (semi-muchii)).
  - **Vârf** v: coordonatele lui v în $Coordinates(v)$, pointer $IncidentEdge(v)$ spre o muchie orientată care are v ca origine
  - **Față** f: pointer $OuterComponent(f)$ spre o muchie orientată corespunzătoare frontierei externe (pentru fața nemărginită este **nil**); listă $InnerComponents(f)$, care conține, pentru fiecare gol, un pointer către una dintre muchiile orientate de pe frontieră
  - **Muchie orientată** $\vec{e}$: pointer $Origin(\vec{e})$, pointer $Twin(\vec{e})$ pointer $IncidentFace(\vec{e})$, pointer $Next(\vec{e})$, pointer $Prev(\vec{e})$.
- Oricărei subdiviziuni planare $\mathscr{S}$ i se asociază o listă de muchii dublu înlănțuite $\mathcal{D}_{\mathscr{S}}$.
- **Obs.** Explicați cum, folosind pointerii de mai sus: (i) poate fi parcursă frontiera exterioară / interioară a unei fețe (a unui poligon); (ii) pot fi găsite toate semi-muchiile incidente cu un vârf.

Exemplu



Vârf	Coordonate	Incident Edge
v_1	$(0,4)$	$\vec{e}_{12}$
v_2	$(4,4)$	$\vec{e}_{23}$
v_3	$(4,0)$	$\vec{e}_{34}$
v_4	$(0,-2)$	$\vec{e}_{43}$

Față	Outer Component	Inner Component(s)
(exterior) f_1	nil	$\vec{e}_{12}$
(Δ) f_2	$\vec{e}_{21}$	nil

Semi-muchie	Origin	Twin	Incident Face	Next	Prev
$\vec{e}_{12}$	v_1	$\vec{e}_{21}$	f_1	$\vec{e}_{23}$	$\vec{e}_{31}$
$\vec{e}_{23}$	v_2	$\vec{e}_{32}$	f_1	$\vec{e}_{34}$	$\vec{e}_{12}$
... $\vec{e}_{34}$ ...	v_3	$\vec{e}_{43}$ ...	f_2	$\vec{e}_{43}$	$\vec{e}_{23}$