ÚLOHA LIN. PROGRAMOVÁNÍ V ROVINĚ

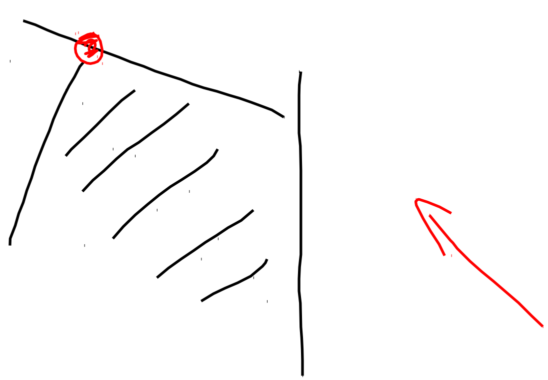
Najít bod, ve kterém nabývá maxima

$$f(x,y) = c_1 x + c_2 y$$

$$a_{i1} x + a_{i2} y \leq b_i \qquad i = 1, 2, \dots n$$

Průnik polorovin

$\vec{c} = (c_1, c_2)$ … hledaný bod je maximální v průniku polorovin ve směru vektoru $\vec{c}$



Řešení omezené úlohy

Za předpokladu, že mezi polorovinami jsou dvě h_1, h_2 tak, že f nabývá maxima na $l_1 \cap l_2$. Pokud je takových bodů více vybereme maximální ve vhodném lexikografickém uspořádání tak, aby to byl bod průniku hraničních přímek – značíme to v_2

$$v_2 = l_1 \cap l_2 \qquad l_i \text{ hraniční poloroviny } h_i$$

Hledáme postupně body maxima v_i v průniku $C_i = \bigcap_{j=1}^{i} h_j$

v_i hledáme na l_i. Lze to najít řešení úlohy 1-dim LP.

l_i: $\quad a_{i1}x + a_{i2}y = b_i$

Předp. $a_{i2} \neq 0 \qquad y = \dfrac{b_i - a_{i1}x}{a_{i2}}$

Toto y dosadíme do $f(x,y)$ a nerovností:

3

$$g(x) = f\left(x, \frac{b_i - a_{i1}x}{a_{i2}}\right) = c_1 x + c_2 \left(\frac{b_i - a_{i1}x}{a_{i2}}\right) = \left(c_1 - c_2 \frac{a_{i1}}{a_{i2}}\right) x + c_2 \frac{b_i}{a_{i2}}$$

$$= kx + m$$

Hledáme maximum funkce kx na množině

$j = 1, 2, \dots i-1$

$$a_{j1} x + a_{j2} \left(b_i - \frac{a_{i1} x}{a_{i2}}\right) \leq b_j$$

$$\left(a_{j1} - a_{j2} \frac{a_{i1}}{a_{i2}}\right) x \leq b_j - a_{j2} b_i$$

Toto je úloha 1. dim. lin. programování

Je-li množina daná nerovnostmi prázdná, eliminuji poloroviny h_j, h_k takové, že $(h_j \cap l_i) \cap (h_k \cap l_i) = \emptyset$.

Potom $h_j \cap h_k \cap h_i = \emptyset$

Časová náročnost

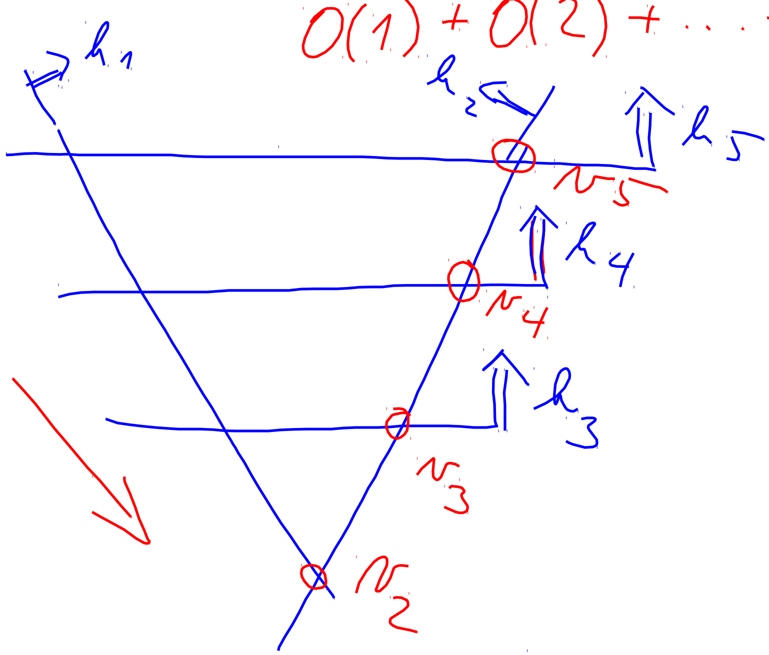
Krok od v_{i-1} k v_i.

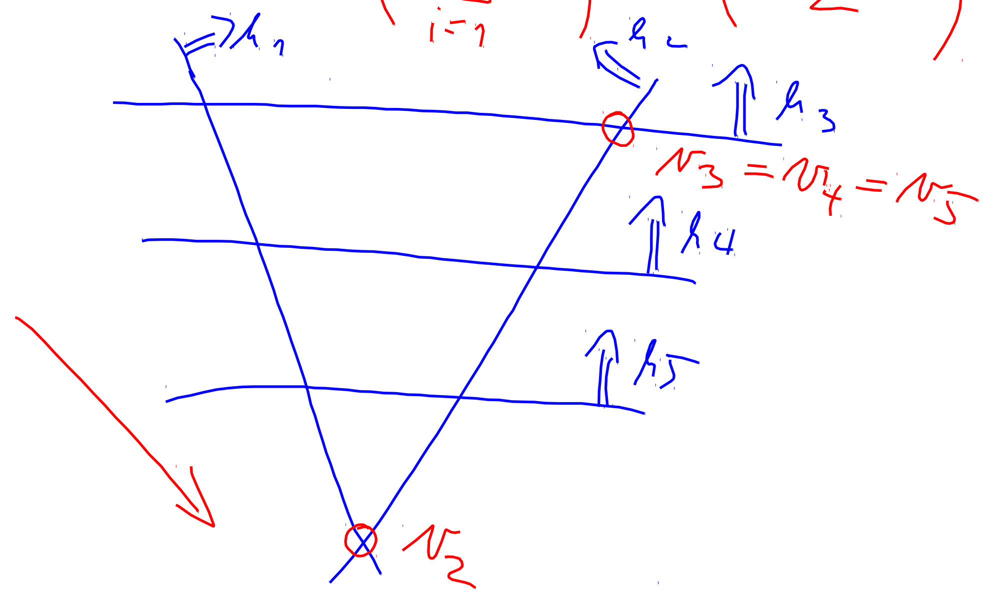
1. $v_{i-1} \in h_i$, pak $v_i = v_{i-1}$ a časová náročnost je konstantní

2. $v_{i-1} \notin h_i$, pak řešíme úlohu 1-dim. lin. programování s časovou náročností $O(i)$.

Tedy celková časová náročnost je nejvýše

$$O(1) + O(2) + \dots + O(n) = O\left(\sum_{i=1}^{n} i\right) = O\left(\frac{(n+1)n}{2}\right) = O(n^2)$$





Časová náročnost závisí na pořadí položení. Očekávaná časová náročnost je průměrná hodnota těchto čas. náročností pro všechna možná uspořádání.

Vezmeme-li tedy pořadí položení náhodné, lze očekávat, že čas vyjde někde blízko této hodnotě.

Odvození očekávané časové náročnosti

$X(i)$ bude náhodná veličina

$$X(i) = \begin{cases} 0 & v_{i-1} \in h_i \\ 1 & v_{i-1} \notin h_i \end{cases}$$

Očekávaná časová náročnost je

$$\sum_{i=3}^{n} \underbrace{E\,X(i)}_{\text{střední hodnota náhodné veličiny}} O(i) + (n-3)\,O(1)$$

Je-li X náhodná veličina nabývající hodnot $a_1, a_2, \dots a_s$, pak její střední hodnota je

$$EX = a_1 \cdot \underbrace{p(X=a_1)}_{\text{pravděpodobnost, že } X=a_1} + a_2 \cdot p(X=a_2) + \dots + a_s \cdot p(X=a_s)$$

V našem případě

$$EX(i) = 0 \cdot p(X(i)=0) + 1 \cdot p(X(i)=1)$$

$$= p(v_{i-1} \notin h_i)$$

v_i leží na přímkách l_j, l_k.

= pravděpodobnost, že jedna z přímek na které leží v_i je l_i.

= pravděpodobnost, že při výběru dvou čísel z $\{3,4,5,\dots i\}$ vybereme i

$$= \frac{2}{i-2}$$

⑦

Dvouprvkové podmnožiny $\qquad \binom{i-2}{2} = \dfrac{(i-2)(i-3)}{2}$

Dvouprvkové podmnožiny obsahující i .... $i-3$

$$p = \frac{i-3}{\frac{(i-2)(i-3)}{2}} = \frac{2}{i-2}$$

Očekávaná náročnost je

$$\sum_{i=3}^{n} E\,X(i)\,O(i) = \sum \frac{2}{i-2}\,O(i) = O\left(\sum_{i=3}^{n} \frac{2i}{i-2}\right)$$

$$= O\left(\sum_{i=3}^{n} 6\right) = O(n)$$

Při náhodném pořadí hledání je očekávaná časová náročnost $O(n)$.

Neomezená úloha lin. programování

Může se stát, že v průniku polorovin leží polopřímka

$$\rho = \{ p + \lambda \vec{d} , \ \lambda \geq 0 \}$$

na které f roste. V tomto případě chceme nějakou takovou polopřímku najít. Při hledání můžeme dojít k výsledkům:

(1) najdeme polopřímku ρ

(2) zjistíme, že $\bigcap_{i=1}^{n} h_i = \emptyset$

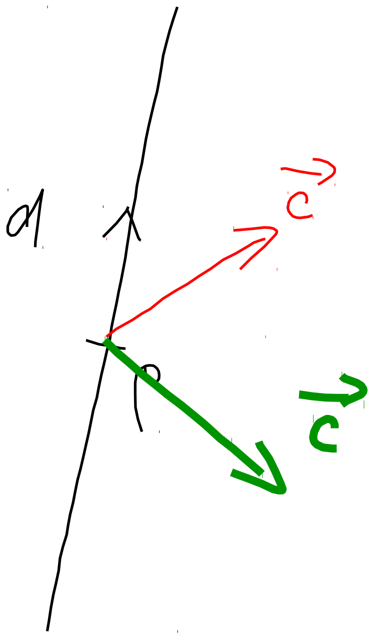
(3) najdeme h_j a h_k takové, že f je omezená na $h_j \cap h_k$.

Následuje řešení omezené úlohy LP.

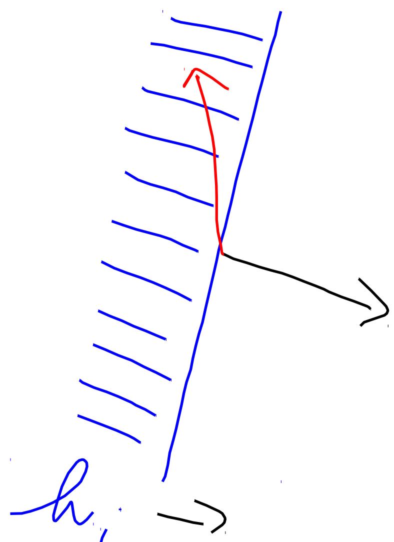
Aby f rostla ve směru vektoru $\vec{d}$ přípustnosti p musí platit

(1) $$\langle \vec{c}, \vec{d} \rangle > 0$$

$\vec{c}$ svírá s $\vec{d}$ ostrý úhel



(2) $\vec{d}$ musí mířit do průniku všech polorovin



$$\langle \vec{d}, \eta_i \rangle \leq 0$$

η_i vnější normála hraniční přímky

Algoritmické hledání vektoru $\vec{d}$

Stačí hledat $\vec{d}$ až na kladný násobek

$$\vec{d} = \vec{c} + t\vec{e}$$ kde $\vec{e}$ je vektor kolmý k $\vec{c}$

Platí

$$\langle \vec{d}, \vec{c} \rangle = \langle \vec{c} + t\vec{e}, \vec{c} \rangle = \underbrace{\langle \vec{c}, \vec{c} \rangle}_{>0} + t \underbrace{\langle \vec{e}, \vec{c} \rangle}_{=0} > 0.$$

Další normály jsou

$$0 \geq \langle \vec{d}, \eta_i \rangle = \langle \vec{c} + t\vec{e}, \vec{\eta_i} \rangle = \langle \vec{c}, \vec{\eta_i} \rangle + t \langle \vec{e}, \eta_i \rangle$$

$h_i: \quad a_{i1}x + a_{i2}y \leq b_i$

$\eta_i = (a_{i1}, a_{i2})$

Toto dává úlohu 1-dim lin. programování pro neznámou t

má řešení t
tak vezmeme
$d = \vec{c} + t\vec{e}$

nemá řešení
protože nějaké dvě
polopřímky určené
η_j a η_k nemají
průnik.

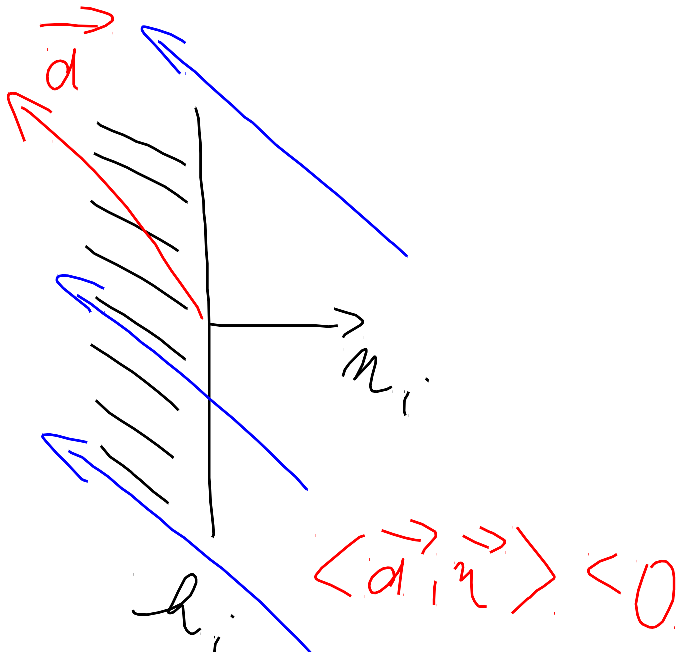
To znamená, že f je omezená
na $h_j \cap h_k$.

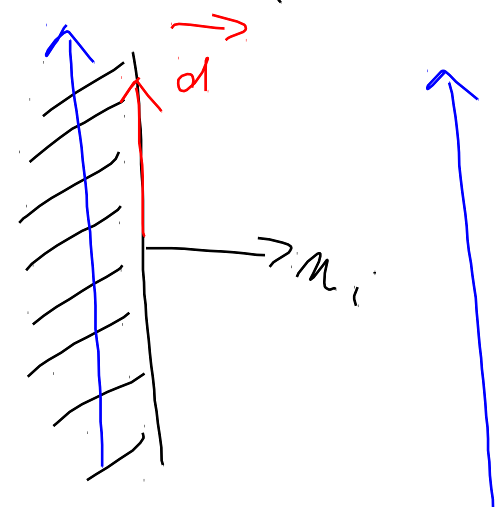
Máme-li $\vec{d}$, které splňuje

$$\langle \vec{d}, \vec{c} \rangle > 0 \quad \text{a} \quad \langle \vec{d}, \vec{n}_i \rangle \leq 0$$

musíme pro existenci polopřímky p zjistit, jak je to s příslušnými poloprostory h_i, pro které

$$\langle \vec{d}, n_i \rangle = 0 .$$



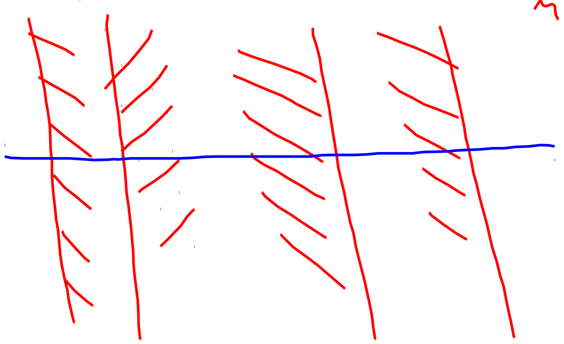


Musíme zjistit, zda $\bigcap_{h_i \in H'} h_i$ kde $H' = \{ h_i \in H, \langle \vec{d}, n_i \rangle = 0 \}$ je prázdný nebo neprázdný!

1. je neprázdný – pak existuje polopřímka se směrovým vektorem $\vec{d}$ ležící v $\bigcap_{h_i \in H'} h_i$ a současně ležící v $\bigcap_{h_i \in H} h_i$

2. Průnik $\bigcap_{h_i \in H'} h_i$ je prázdný. Pak je prázdný i průnik všech polorovin.

Jak zjistíme, že $\bigcap_{h_i \in H'} h_i$ je prázdný nebo neprázdný.



Vezmeme l kolmou na hraniční přímky těchto polorovin a řešíme úlohu 1-dim. lin. programování.

Ortogonální vyhledávání

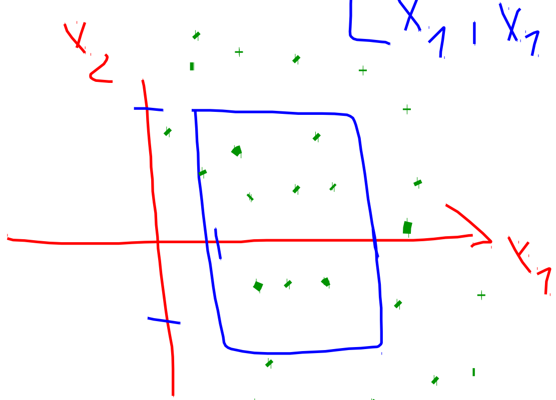
Máme databázi a každá položka má několik číselných údajů.

Zaměstnanci firmy – $\underbrace{\text{věk, jak dlouho pracují, výše mzdy, } \dots}_{d\text{-údajů}}$

Geometricky: množina bodů v $\mathbb{R}^d$

Jde o to vytvořit pro danou množinu bodů (zaměstnanců) vyhledávací strukturu, která umožní rychle zjistit, které body leží v kvádru

$$[x_1, x_1'] \times [x_2, x_2'] \times \dots \times [x_d, x_d']$$



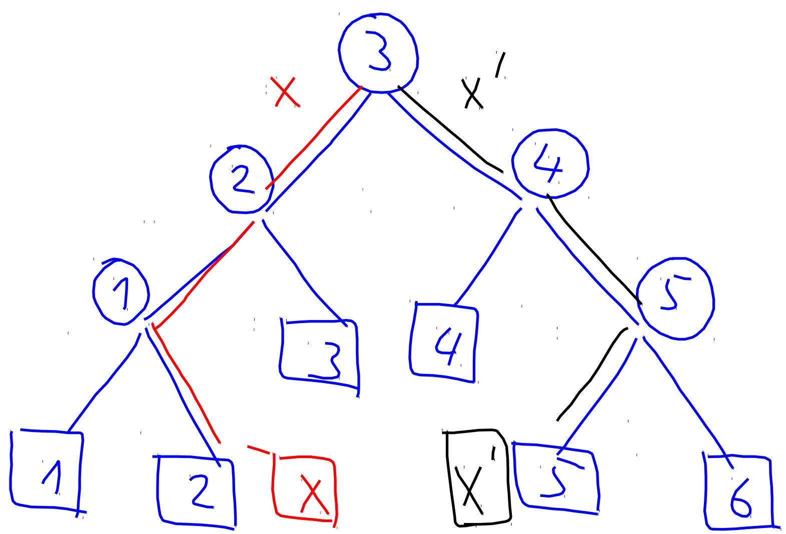
Dvě struktury

kd-stromy   kd-trees

range trees

Dimenze 1 – obě struktury zde splývají.

Máme množinu n čísel ... požadovaná struktura bude binární vyvážený strom v jehož listech jsou tato čísla uspořádaná podle velikosti:



Pro zadaný interval $[x, x']$, $x \leq x'$ hledáme, která čísla v něm leží.

Časová náročnost je

$O(\log n + k)$, kde k je počet čísel v intervalu $[x, x']$.

Dvě „vylepšení"

Cesta ve stromu pro x a x' může být zpočátku společná.

Uzel stromu, kde se cesty rozcházejí nazveme štěpící uzel (split node) a ten najdeme prvně.