Kombinatorická optimalizace

přednáška 19.2.2019

- na přednášce č. 6 se bude psát test, 2 příklady na lin prog v $\mathbb{Z}$, jeden test bude někdy později ← nelze opravit známku
- na cvičeních se bude odevzdávat 5 úkolů, body z nich jsou cenné
- na cvičeních se bude psát jeden programovací test ←

Optimalizace: minimalizujeme / max. fci za nějakých podmínek
kombinatorická = úkoly obsahují diskrétní proměnné

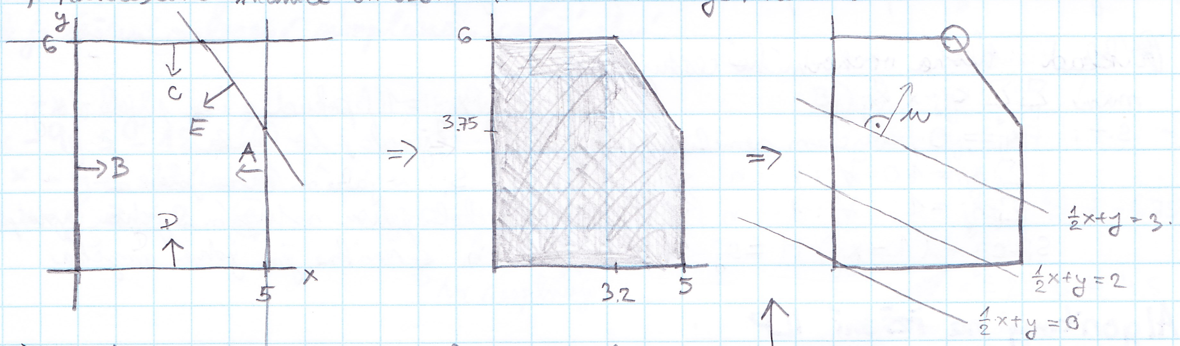
Lineární programování je optimalizační úloha $\max \{ \underbrace{c^T x}_{\text{kriteriální fce}} \mid \underbrace{Ax \le b}_{\text{omezení}} \}$ nebo min

cvičení 21.2.2019

Lin. program může mít:
- 0 řešení: podmínky (omezení) jsou nesplnitelné
- 1 řešení: řešení leží na vrcholu mnohostěnu
- ∞ řešení: řešení utíká do ∞ na neomezené množině, nebo leží na hraně mnohostěnu

Př: Řešte $\max\{\frac{1}{2}x + y \mid \overset{A}{x \le 5}, \overset{B}{x \ge 0}, \overset{C}{y \le 6}, \overset{D}{y \ge 0}, \overset{E}{y \le -\frac{5}{4}x + 10}\}$

1) Nakreslím hranice omezení (množinu možných řešení)



2) Nakreslím vrstevnice kriteriální fce f, nejlépe se to dělá dosazením vhodných dvou bodů. Vrstevnice budou rovnoběžné

3) Hodnota kriteriální fce roste ve směru vektoru $\vec{u}$. Maximální hodnoty bude nabývat v tomto vrcholu, kde platí:

$y = 6$

$y = -\frac{5}{4}x + 10$

$-4 = -\frac{5x}{4}$

$-16 = -5x$

$x = 16/5 = 3.2$

optimální řešení: $[x = 3.2,\ y = 6]$

$f(3.2, 6) = \frac{3.2}{2} + 6 = 1.6 + 6 = 7.6$

optimální hodnota: $\underline{7.6}$

přednáška 26.2.2019

Integer linear programming – ILP

$$\begin{array}{c} \\ m \end{array}\!\!\begin{array}{c} n \\ \begin{bmatrix} A \end{bmatrix} \end{array} \cdot \begin{array}{c} 1 \\ \begin{bmatrix} x \end{bmatrix} \end{array}\!\! n \le \begin{array}{c} 1 \\ \begin{bmatrix} b \end{bmatrix} \end{array}\!\! m$$

, hledáme max $c^T x$, kde $x \in \mathbb{Z}^n$ — stavový prostor je nekonvexní

problém je lineární, protože vznikne soustava lin. nerovnic a maximalizujeme lineární funkci. Prostor vytvořený lineárními nerovnicemi bude konvexní (v $\mathbb{R}^n$)

Rozdíl mezi LP a ILP
- prostor řešení ILP není konvexní ⇒ řešení je náročnější
- najít řešení v LP a zaokrouhlit ho je velký problém, musí se na to jinak
- ILP je NP úplný problém

Příklad: Dělení lupu - bankovek co nejrovnějším dílem mezi dvě lupiče.

min 0

s.t.: $\sum x_i \cdot p_i = \sum (1-x_i)\cdot p_i$

$p = [100, 50, 50, 50, 20, 20, 10, 10]$

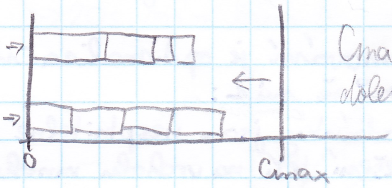
$x = [1, 1, 0.1, 0, 0, 0, 0, 0]$ ← najít řešení je snazší, pokud povolíme dělení bankovek

Příklad: Chceme minimalizovat rozdíl mezi výdělky dvou lupičů

min C_{max}

s.t.: $\sum x_i p_i \le C_{max}$ ← lupič 1

$\sum (x_i - 1)\, p_i \le C_{max}$ ← lupič 2



Cmax tlačíme co nejvíc doleva

Příklad: Hledání nejkratší cesty

max l_1

s.t.: $l_s = 0$

$l_j \le l_i + c_{ij}$, $i \in 1..n$, $j \in 1..n$

Nejkratší cesta povede po hranách, které nejsou zkrácené.

Příklad: Úloha obchodního cestujícího

min $\sum_i \sum_j c_{ij} \times x_{ij}$

$x_{ij} = 1$, pokud z i se půjde do j, jinak je to nula

s.t.: $x_{ii} = 0$, abychom neměli smyčky

$\sum_i x_i = 1$ $\forall j: 1..n$ ←

s_i: v jakém kroku jdeme do i

$\sum_j x_j = 1$ $\forall i: 1..n$ ← do každého vrcholu jednou vstoupím a jednou vystoupím

$s_i + c_{ij} - (1 - x_{ij})\cdot M \le s_j$ $\forall i \in 1..n, j \in 2..n$ ← vznikne jen jedna smyčka

Algoritmy na řešení ILP

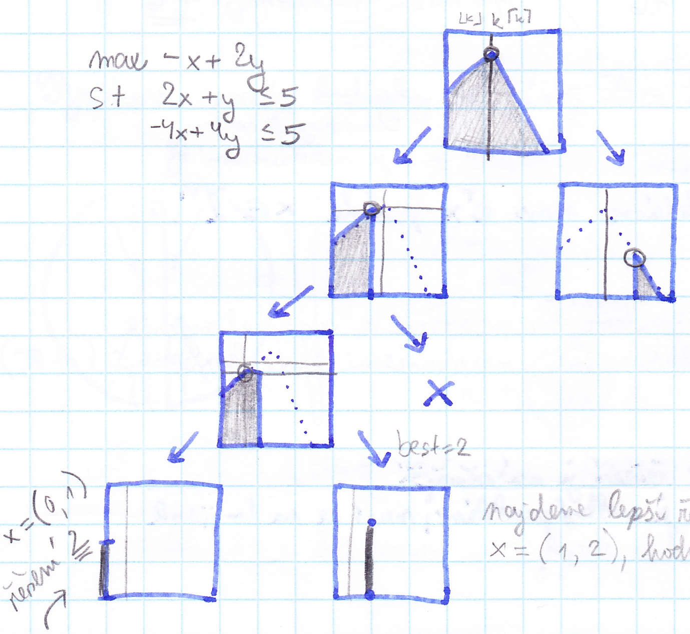
Enumerační metody: najdeme všechna možná řešení (jejich výčet) a vybereme optimum. Na malých vstupech fungují dobře.

Metoda větví a mezí (branch and bound): Pomocí LP najdeme nejlepší hodnotu kritéria, to bude v bodě k. Vybereme nějakou neceločíselnou souřadnici k_i a podle ní rozdělíme úlohu na dva podproblémy přidáním $\lceil k_i \rceil$ a $\lfloor k_i \rfloor$, mezi kterými žádné celočíselné řešení není.
Pokud v podproblému LP najde horší řešení než našlo ILP výše, tak tuhle větev zahodím.

max $-x + 2y$

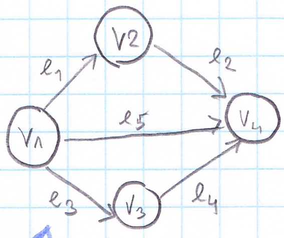
s.t. $2x + y \le 5$

$-4x + 4y \le 5$



vrátili jsme se nahoru, nejlepší je $x = (1,2) = 3$

najdeme lepší řešení $x = (1, 2)$, hodnota 3



$$W = \begin{array}{c|ccccc} & e_1 & e_2 & e_3 & e_4 & e_5 \\ \hline v_1 & 1 & 0 & 1 & 0 & 1 \\ v_2 & -1 & 1 & 0 & 0 & 0 \\ v_3 & 0 & 0 & -1 & 1 & 0 \\ v_4 & 0 & -1 & 0 & -1 & -1 \end{array}$$

$\sum x_j \cdot w_{ij} = 1$, $i = 1$

Z vrcholu i odchází pouze jedna hrana

$x_j = 1$ iff e_j je na cestě (je vybraná)

jinak $x_j = 0$

$\sum x_j \cdot w_{ij} = -1$, $i = 4$

Do koncového vrcholu vchází jen jedna hrana

Hledání nejkratší cesty v grafu jako ILP

$\sum x_j \cdot w_{ij} = 0$, $i \neq 1$, $i = 4$

Definice: Matice A je unimodulární, pokud všechny její čtvercové podmatice mají determinant 0 nebo ±1.

Lemma: Nechť $A \in \mathbb{R}^{m \times n}$ je unimodulární matice a $b \in \mathbb{Z}^n$. Pak každý vrchol mnohostěnu $P = \{x;\ Ax \le b\}$ je celočíselný.

Lemma: Pokud je ILP zadané unimodulární maticí A a celočíselným vektorem b, pak je řešení nalezené simplexem celočíselné.

Cvičení 28.2.2019

min $-x + 2y$

$-4x - 9y \le -18$ (A)

$3/2\, x - y \le 27/4$ (B)

$8/17\, x - y \ge -2$ (C)

A) $-4x - 9y \le -18$

$x = 0$: $-9y \le -18$ /:9

$y \ge 2$

$y = 0$: $-4x \le -18$ /:(-2)

$\cancel{2}x \ge \cancel{9}\ 4.5$

P to nesplňuje

B) $\frac{3}{2}x - y \le \frac{27}{4}$ /·4

$6x - 4y \le 27$

$x = 0$: $-4y \le 27$

$y \ge -\frac{27}{4} \doteq -6.75$

$y = 0$: $6x \le 27$

$x \le 27/6 \doteq 4.5$

P to splňuje

c) $\frac{8}{17}x - y \ge -2$ /·17

$8x - 17y \ge -34$

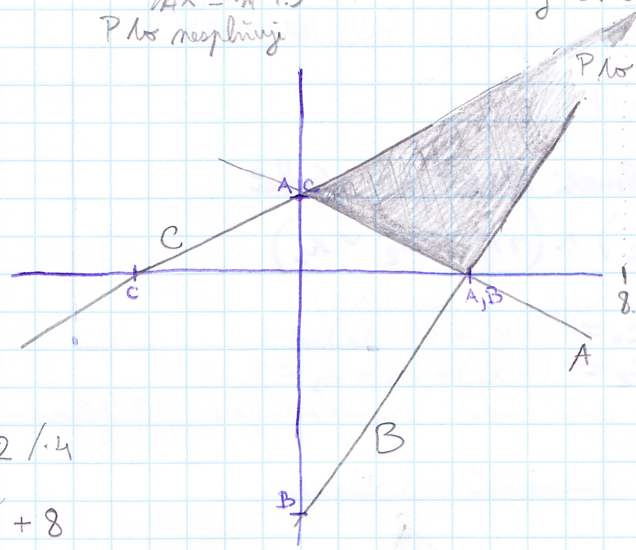
$x = 0$: $-17y \ge -34$

$y \le 2$

$y = 0$: $8x \ge -34$

$x \ge -\frac{34}{8} \doteq -4.25$

P to splňuje



$B = C$

$\frac{3}{2}x - y - \frac{27}{4} = \frac{8}{17}x - y + 2$ /·4

$6x - \cancel{4y} - 27 = \frac{32}{17}x - \cancel{4y} + 8$

$6x - \frac{32}{17}x = 35$ /·17

$102x - 32x = 595$

$70x = 595$

$x = 8.5$

$y = \frac{3}{2}x - \frac{27}{4} = \frac{51 - 27}{4} = \frac{24}{4} = 6$

Úloha: minimální vrcholové pokrytí

$G = (V, E)$

$V^* \subseteq V$, $\forall e = \{u, v\} : u \in V^* \vee v \in V^*$

1) proměnné: x_v, $\forall v \in V : x_v \in \{0, 1\}$

2) omezení: $\forall e = \{u, v\} : x_u + x_v \geq 1$

3) kritérium: $\min \sum_{v=1}^{|V|} x_v$

Formulace v Gurobi:

```
import gurobipy as g

n=6
edges = [(1,2),

model = g.Model()

x = model.addVars(n, vtype = g.GRB.BINARY)

for u,v in edges:
  model.addConstr(x[u] + x[v] ≥ 1)

expr = 0
for idx in range(n)
  expr += x[idx]
model.setObjective(expr, g.GRB.MAXIMIZE)

model.optimize()
```

Řešení splnitelnosti logických výroků

$\varphi = (x_1 \vee \neg x_2) \wedge (\neg x_1 \vee x_2 \vee x_3)$

1) proměnné: $x_1, x_2, x_3 \in \{0, 1\}$

2) omezení: $c_1 = x_1 + (1 - x_2) \geq 1$

$c_2 = (1 - x_1) + x_2 + x_3 \geq 1$

3) kritérium: není nutné, zajímá nás pouze splnitelnost