Poznámky ke státnicím

Cyril Kotecký

Chapter 1

Úvod do složitosti a vyčíslitelnosti

1.1 Výpočetní modely (Turingovy stroje, RAM)

TM

Definition: ***TM***

$$M = (Q, \Sigma, \delta, q_0, F)$$

- Q konečná množina stavů
- Σ konečná abeceda
- $\delta : Q \times \Sigma \to Q \times \Sigma \times \{R, N, L\} \cup \perp$ přechodová funkce ($\perp$ nedefinovaný)
- $q_0 \in Q$ počáteční stav
- $F \subseteq Q$ přijímající stavy

1. **Počáteční stav:** Hlava na levém symbolu vstupního slova na pásce.
2. **Krok:** Podle δ změníme stav, přepíšeme znak a posuneme hlavu.
3. **Konec:** Končíme když je přechod nedefinovaný, přijímáme pokud je stav v F .

Definition: ***Jazyky TM***

- $L(M)$ jazyk přijímavý M
- $M(w) \downarrow$ výpočet na w skončí
- $M(w) \uparrow$ výpočet na w neskončí
- L je TM **částečně rozhodnutelný/rekurzivně spočetný** , pokud

$$\exists M : \ L = L(M) \ .$$

- L je TM **rozhodnutelný/rekurzivní** , pokud

$$\exists M : \ L = L(M) \ \wedge \ \forall w \in \Sigma^* : \ M(w) \downarrow .$$

Definition: ***Funkce TM***

M počítá částečnou funkci $f_M : \Sigma^* \to \Sigma^*$, kde $f(w) \downarrow$ pokud $M(w) \downarrow$.
f je TM vyčíslitelná pokud $\exists M$, který jí počítá.

Example: ***Převod vícepáskového na jednopáskový***

- $\Sigma' := \Sigma \cup \{H, -\} \times \Sigma^k$
- Pásky zarovnáme podle pozic hlav a dorovnáváme po posunu hlav.

Example: ***Převod oboustranného na jednostranný***

Example: ***Převod vícehlavého na jednohlavého***

UTM

Definition: *UTM*

1. **UTM** : $\mathcal{U}$ na vstupu $\langle M, x\rangle$ simuluje práci M na x .
2. **LU:** $L_{\mathcal{U}} := \{\langle M, x\rangle |\ x \in L(M)\}$
3. **DIAG:** DIAG $:= \{\langle M\rangle |\ \langle M\rangle \notin L(M)\}$

Theorem: *Částečná rozhodnutelnost LU*

1. $L_{\mathcal{U}} \in$ PD
2. DIAG $\notin$ PD
3. $L_{\mathcal{U}} \notin$ DEC

Proof:

1. Z existence $\mathcal{U}$.
2. Jinak $\exists M_D$:
   - $\langle M_D\rangle \in$ DIAG $\iff \langle M_D\rangle \notin L(M_D)$ z definice DIAG.
   - $\langle M_D\rangle \in$ DIAG $\iff \langle M_D\rangle \in L(M_D)$ z DIAG definice $L(M_D)$.
3. Jinak $\exists M_{\mathcal{U}}$ a lze ho použít pro rozhodování DIAG

$$\forall M :\ \langle M\rangle \in \text{DIAG} \iff > \langle M\rangle \notin L(M) \iff \langle M, \langle M\rangle\rangle \notin L_{\mathcal{U}} .$$

RAM

Definition: *RAM*

- Registry $\{r_i\}_{i\in\mathbb{N}}$ kde $r_i \in \mathbb{N}$ a inicializované na 0 .
  - $[r_i]$ obsah registru
  - $[[r_i]] := [r_{[r_i]}]$
- Program $P = I_0, ..., I_l$ konečná posloupnost instrukcí.
  - LOAD, ADD, SUB, COPY, JNZ, READ, PRINT
- Pole $A_1, ...A_p$ uložené jako $A_i[j] = r_{i+j(p+1)}$.
- Proměnné uložené jako $r_{i(p+1)}$.

Definition: *Jazyky RAM*

- R čte slovo $w = \sigma_{i_1}...\sigma_{i_n}$ jako posloupnost $i_1, ..., i_n$ zakončenou 0 .
- R přijme w pokud $R(w)\downarrow$ a R jako první napíše 1 .
- R odmítne w pokud $R(w)\downarrow$ a R napíše nic nebo nezačne 1 .

1. L je RAM **částečně rozhodnutelný** pokud $\exists R :\ L = L(R)$.
2. L je RAM **rozhodnutelný** pokud $\exists R :\ L = L(R) \ \wedge\ \forall w \in \Sigma^* :\ R(w)\downarrow$.

Definition: *Funkce RAM*

R počítá částečnou aritmetickou $f : \mathbb{N}^n \to \mathbb{N}$ pokud pro vstup $x = x_1, ..., x_n$:

- $f(x)\downarrow \implies R(x)\downarrow$ a R vypíše $f(x)$
- $f(x)\uparrow \implies R(x)\uparrow$

R počítá částečnou $f : \Sigma^* \to \Sigma$, pokud pro vstup $w = \sigma_{i_1}...\sigma_{i_n}$ daný jako $i_1, ..., i_n, 0$:

- $f(w)\downarrow = \sigma_{j_1}...\sigma_{j_m} \implies R(w)\downarrow$ a R vypíše $j_1, ...j_m, 0$
- $f(w)\uparrow \implies R(w)\uparrow$

Převody

Algorithm: *R*

1. Jednostrannou pásku načteme do pole T ukončeného 0 .
2. $q = 0,\ h = 0$
3. Dokud $\delta(q, T[h]) \neq \perp$: aktualizujeme $T[h], h, q$ podle δ reprezentované jako posloupnost podmínek.
4. Pokud $q \in F$, vypíšeme 1 , jinak 0 .
5. Vypíšeme obsah pásky.

Algorithm: *M*

- **Vstupní páska a výstupní páska:** Čísla zapsaná binárně, oddělená znakem # .
- **RAM páska:** Registry $r_{i_1}, ..., r_{i_m}$ pro $i_1 < ... < i_m$ reprezentujeme jako

$$(i_1)_B|([r_{i_1}])_B\#(i_2)_B|([r_{i_2}])_B\#...\#(i_m)_B|([r_{i_m}])_B \ .$$

- **Pomocná páska:** Součty, rozdíly, nepřímá adresace, posuny etc.
- Číslo instrukce si pamatujeme ve stavu.
- Jedna instrukce R je posloupnost instrukcí M .
  1. Lokace registrů s operandy.
  2. Aritmetika s operandy.
  3. Výpočet adres.
  4. Úprava paměti (nový registr, úprava registr, posun pásky s pamětí).
- Výpis 1 jako první si pamatujeme ve stavu.
- Končíme po poslední instrukci.

1.2 Základní třídy složitosti a jejich vztahy

Základní třídy

Definition: *Složitost TM*

- M **pracuje v čase** $f(n)$ pokud $\forall x \in \Sigma^n,\ |x| = n :\ M$ na x končí po $\leq f(n)$ krocích.
- M **pracuje v prostoru** $f(n)$ pokud $\forall x \in \Sigma^n,\ |x| = n :\ M$ na x skončí a použije $\leq f(n)$ buněk pásky.
- N **pracuje v čase** $f(n)$ pokud $\forall x \in \Sigma^n$ a $\forall$ výpočet M na x končí po $\leq f(n)$ krocích.
- N **pracuje v prostoru** $f(n)$ pokud $\forall x \in \Sigma^n$ a $\forall$ výpočet M na x končí a použije $\leq f(n)$ buněk pásky.

Definition: *Třídy složitosti*

- TIME $(f(n))$ jazyky přijímané TM v čase $O(f(n))$.
- SPACE $(f(n))$ jazyky přijímané TM v prostoru $O(f(n))$.
- NTIME $(f(n))$ jazyky přijmané NTM v čase $O(f(n))$.
- NSPACE $(f(n))$ jazyky přijímané NTM v prostoru $O(f(n))$.

Observation: *Inkluze*

1. TIME $(f(n)) \subseteq$ SPACE $(f(n))$
2. TIME $(f(n)) \subseteq$ NTIME $(f(n))$
3. SPACE $(f(n)) \subseteq$ NSPACE $(f(n))$
4. L $\subseteq$ NL $\subseteq$ P $\subseteq$ NP $\subseteq$ PSPACE $\subseteq$ EXPTIME $\subseteq$ NEXPTIME $\subseteq$ EXPSPACE

Theorem:

$$\text{NTIME}\ (f(n)) \subseteq \text{SPACE}\ (f(n))$$

Proof: Postupně prodlužujeme větve NTM podle kterých simulujeme, dokud NTM nepřijme. Jinak odmítneme.

Corollary:

$$\text{NP} \subseteq \text{PSPACE}$$

Theorem: ***Savičova věta***

$$f(n) \geq \log_2 n \implies \text{NPSPACE}\ (f(n)) \subseteq \text{SPACE}\ (f^2(n))\ .$$

Proof:

Algorithm: M'

1. $i = 1$
2. Pokud Reachable $(C_0^x,\ C_{\text{acc}},\ c_M i)$:
3. $\forall$ konfiguraci C s prostorem $i+1$:
4. Pokud Reachable $(C_0,^x, C,\ c_i)$:
5. $i = i + 1$
6. Zpět na 2.
7. Odmítni

Algorithm: ***Reachable***

- Prostor $O(f(n))$ na konfiguraci a $O(f(n))$ na k
- Nejvýš $O(f(n))$ instancí

1. Pokud $k = 0$
2. Pokud $C_1 = C_2$ nebo $C_1, C_2 \in E$, odpověď **ano**
3. Jinak **ne**
4. $\forall$ konfiguraci C_m v prostoru $f(n)$:
6. Pokud Reachable $(C_1, C_m, k-1)$ a Reachable $(C_m, C_2, k-1)$, odpověď **ano**
7. Odpověď **ne**

Corollary:

$$\text{NSPACE} = \text{PSPACE}$$

Corollary:

- NL $\subseteq$ P
- NSPACE $\subseteq$ EXPTIME

NP

Definition: ***Verifikátor***

$$V \text{ pro } L \text{ je } V \text{ splňující } L = \{x|\ \exists y :\ V(x,y) \text{ přijme } \}\ :$$

- Pracuje v čase $n^{O(1)}$.
- Předpokládáme $y \in n^{O(1)}$.

Definition: ***NP***

Jazyky s polynomiálními verifikátory.

Theorem: ***Alternativní definice***

$$\text{NP} = \bigcup_{k\in\mathbb{N}} \text{NTIME}\ (n^k)$$

Proof:

$\subseteq$ Nedeterministicky volíme znaky důkazu y .

$\supseteq$ Posloupnost větvení k přijetí dávají certifikát.

Theorem: ***Cook-Levin***

SAT je NP-úplný.

Proof:

- NP: Splňující ohodnocení je certifikát.
- Úplnost:
  - Máme NTM pracující v čase n^k s jediným přijímajícím q_1 .
  - Řádky délky n^k jsou konfigurace M , stav zapsán před pozicí hlavy.
  - Řádky v tableau velikosti $n^k \times n^k$ následují podle přechodové funkce. Přechody jsou platné právě když jsou platná všechna okna 3×2 . Přípustná okna P .
  - Přijímající tableau obsahuje řádek s přijímající konfigurací.
  - Symboly tableau $S = Q \cup \Sigma \cup \{\#\}$. Proměnné $x_{i,j,s} = 1 \iff T[i,j] = s$.
  - Formule $\varphi = \varphi_{\text{cell}} \wedge \varphi_{\text{start}} \wedge \varphi_{\text{move}\wedge\varphi_{\text{accept}}}$.

$$\varphi_{\text{cell}} = \bigwedge_{i,j} \left(\left(\bigvee_s x_{i,j,s} \right) \ \wedge \ \left(\bigwedge_{s\neq t} (\neg x_{i,j,s} \vee \neg x_{i,j,t}) \right) \right)$$

$$\varphi_{\text{move}} = \bigwedge_{i,j} \varphi_{\text{legal}_{i,j}}$$

$$\varphi_{\text{accept}} = \bigvee_{i,j} x_{i,j,q_1}$$

$$\varphi_{\text{legal}_{i,j}} = \bigvee_{(s_1,\ldots,s_6)\in P} (x_{i,j,s_1} \wedge x_{i,j+1,s_2} \wedge x_{i,j+2,s_3} \wedge x_{i+1,j,s_4} \wedge x_{i+1,j+1,s_5} \wedge x_{i+1,j+2,s_6})$$

1.3 Aproximační algoritmy a schémata

Aproxmační algoritmy

Definition: ***Optimalizační problém***

$$(\mathcal{I}, \mathcal{F}, f, g)$$

- $\mathcal{I}$ **:** množina vstupů
- $\mathcal{F}$ **:** $\forall I \in \mathcal{I}:\ F(I)$ přípustná řešení
- f **:** $\forall I \in \mathcal{I}\ \forall S \in F(I):\ f(I,S) \to \mathbb{R}$ účelová funkce
- g **:** min / max

Definition: ***Aproximační algoritmus***

A je α -aproximační pro optimalizační problém pokud:

- Spočítá $S \in F(I)$ v polynomiálním čase
- $\forall I \in \mathcal{I}:\ f(I,S) \leq R\,\text{OPT}(I)$ pro minimalizaci

Algorithm: ***Christofides***

1. W liché vrcholy MST T .
2. M minimální perfektní párování na W .
3. Vrátíme průchod $T \uplus M$.

Observation: ***Christofides analýza***

- Handshaking lemma $\implies$ perfektní párování existuje.
- $d(T) \leq \mathrm{OPT},\ d(M_1) + d(M_2) \leq \mathrm{OPT}$

Algorithm: ***Rand-SAT***

1. Nastavíme proměnné uniformně náhodně.

Observation: ***Rand-SAT analýza***

- $P[Y_j = 1] = 1 - 2^{-l} \geq \frac{1}{2}$
- $\mathrm{OPT} \leq \sum w_j \implies \frac{1}{2}\,\mathrm{OPT} \leq \sum w_j \mathbb{E}[Y_j] = \mathbb{E}[W]$

Algorithm: ***Max-Cut***

1. Vyřešíme SDP

$$\max \tfrac{1}{2} \textstyle\sum_E w_{i,j}(1 - V_{i,j}) :\ V_{i,i} = 1,\ V \succeq 0\ .$$

2. Vybereme uniformně náhodný $r \in S^{n-1}$.
3. Vrátíme $\{i|\ v_i r \geq 0\}$.

Observation: ***Max-Cut analýza***

- $\forall x \in (0, \pi) :\ \frac{x}{\pi} \geq 0.878 \frac{1-\cos x}{2}$
- $P[i, j \text{ rozdělené }] = \frac{\theta_{i,j}}{\pi}$
- $\mathbb{E}[W] = \sum \frac{\theta_{i,j}}{\pi} \geq 0.878 \sum \frac{1-\cos\theta_{i,j}}{2} = 0.878 \sum \frac{1-V_{i,j}}{2}$.

Aproximační schémata

Definition: ***(FPTAS)***

- **PTAS:** Množina $(1-\epsilon)$ -aproximačních algoritmů pro všechna $\epsilon > 0$.
- **FPTAS:** PTAS kde každý algoritmus běží v čase $p(|I|, \frac{1}{\epsilon})$.

Algorithm: ***Knapsack pseudopoly***

- Tabulka s řádky $0 \leq k \leq n$ a sloupci $0 \leq c \leq C = \sum c_i$.
- $A_k(c)$ minimální hmotnost batohu s předměty z $[k]$ s cenou c .

$$A_k(c) = \min \begin{cases} h_k + A_{k-1}(c - c_k) & \text{pokud } c \geq c_k \\ A_{k-1}(c) \end{cases}$$
$$A_0(0) = 0,\ A_0(1) = \ldots = A_0(C) = \infty$$

- $B_k(c)$ může ukládat poslední přidaný předmět.
- $c^* := \max\{c :\ A_k(c) \leq H\}$
- $O(nC)$

Algorithm: ***Knapsack FPTAS***

1. Odstraníme předměty těžší než H
2. $c_{\max} = \max c_i$
3. $M := \lceil \frac{n}{\epsilon} \rceil$
4. $\forall i \in [n] :\ \tilde{c}_i = \lfloor c_i \frac{M}{c_{\max}} \rfloor$

5. Vyřešíme dynamický program

Observation: ***Analýza***

$$\tilde{C} \leq nM \in O\left(\frac{n^2}{\epsilon}\right) \implies O(n\tilde{C}) \in O\left(\frac{n^3}{\epsilon}\right)$$

- $\tilde{c}(\mathrm{OPT}) = \sum_{i \in \mathrm{OPT}} \tilde{c}_i = \sum_{i \in \mathrm{OPT}} \left\lfloor c_i \frac{M}{c_{\max}} \right\rfloor \geq \left(\sum_{i \in \mathrm{OPT}} c_i \frac{M}{c_{\max}} \right) - n = c(\mathrm{OPT}) \frac{M}{c_{\max}} - n$
- $c(A) = \sum_{i \in A} c_i \geq \sum_{i \in A} \tilde{c}_i \frac{c_{\max}}{M} = \tilde{c}(A) \frac{c_{\max}}{M} \geq \tilde{c}(\mathrm{OPT}) \frac{c_{\max}}{M}$
- $c(A) \geq \left(c(\mathrm{OPT}) \frac{M}{c_{\max}} - n \right) \frac{c_{\max}}{M} \geq c(\mathrm{OPT}) - \frac{n c_{\max}}{\frac{n}{\epsilon}} \geq c(\mathrm{OPT}) - \epsilon c_{\max} \geq (1 - \epsilon) c_{\mathrm{OPT}}.$

Chapter 2

Datové struktury

2.1 Vyhledávací stromy ((a,b)-stromy, splay stromy).

(a, b) -stromy

Definition: $(a, b$ ***-strom)***

$$2 \leq a,\ 2a - 1 \leq b$$

- Vnitřní vrcholy mají a až b potomků.
- Kořen má 2 až b potomků.
- Listy na stejné hladině.

Lemma:

$$\text{Hloubka s } n \text{ klíči je } \Omega(\log_b n),\ O(\log_a n)\ .$$

Proof:

- Minimální počty vrcholů po hladinách $0, 1, 2a, 2a^2, \ldots$, minimální počty klíčů v hladinách $1, a-1, a-1, \ldots$
- A_h minimální počet klíčů na hloubku h . Inverzní funkce dává maximální hloubku. Minimum analogicky.

$$A_h = 1 + \textstyle\sum_{i \in [h-1]} 2a^{i-1}(a-1) = 2a^{h-1} - 1$$

Definition: ***Operace***

- **Find:** $O(\log_a n \log_2 b) = O(\log n \frac{\log b}{\log a}) \in O(\log n)$
- **Insert:** $\theta(b \log_a n)$
  - Vkládáme na nejhlubší nalezenou vnitřní hladinu.
  - Pokud přeteče, štěpíme. Prostřední klíč dáme do rodiče a vrchol rozdělíme. Případně rekurzíme.
- **Delete:** $\theta(b \log_a n)$
  - Převedeme na mazání z nejhlubší vnitřní hladiny nahrazením následníkem.
  - Pokud podteče, slučujeme. Pokud máme malého sourozence, sloučíme se s ním a s klíčem v rodiči. Pokud máme velké sourozence, vezmeme klíč z rodiče a do rodiče dáme klíč ze sourozence.

Theorem:

$$m \text{ insertů do prázdného } (a, b) \text{ -stromu změní } O(m) \text{ vrcholů.}$$

Proof: Počet štěpení je nejvýš takový jako počet vrcholů na konci.

Theorem:

$$m \text{ insertů a deletů do prázdného } (a, 2a) \text{ -stromu změní } O(m) \text{ vrcholů.}$$

Proof:

- $\Phi = \sum_{v \neq r} f(k(v))$, kde $f(k(v))$ je příspěvek vrcholu v s $k(v)$ klíči.
- Zvolíme $f(k) = \begin{cases} 2 & k = a-2 \\ 1 & k = a-1 \\ 0 & ... \\ 2 & k = 2a-1 \\ 4 & k = 2a \end{cases}$ aby:
  - Konstantní změny způsobí konstantní změny: $|f(i) - f(i+1)| \in O(1)$.
  - Slučování zdarma: $f(2a) \geq f(a) + f(a-1) + O(1)$.
  - Štěpení zdarma: $f(a-2) + f(a-1) \geq f(2a-2) + O(1)$.
- Insert přidá klíč $O(1)$ a změní potenciál o $O(1)$. Štěpení amortizovaně zdarma.
- Delete smaže klíč $O(1)$ a půjčí si nejvýš jeden klíč $O(1)$. Slučování amortizovaně zdarma.

Splay stromy

Definition: *Splay operace*

1. **Zig:** $x : (C, y),\ y : (A, B) \ \rightarrow\ x : (A, y),\ y :\ (B, C)$.
2. **Zig-zig:** $x : (A, B),\ y : (x, C),\ z : (y, D) \ \rightarrow\ x : (A, y),\ y : (B, z),\ z : (C, D)$.
3. **Zig-zag:** $x : (B, C),\ y : (A, x),\ z : (y, D) \ \rightarrow\ x : (y, z),\ y : (A, B),\ z :\ (C, D)$

Definition: *Potenciál*

$$\Phi := \textstyle\sum_{u \in V} r(v) \text{ pro } r(v) := \log_2 s(v) \text{ a } s(v) = |T(v)| \ .$$

Definition: *Operace*

- Cestu z kořene naúčtujeme rotacím. Ty se uamortizují na $O(\log n)$.
- **Find:** Splayujeme nejhlubší nalezený vrchol.
- **Insert:**
  - Pokud vrchol neexistuje, přidání listu l změní potenciál na cestě $v_1, .., v_k :\ s'(l) = 1,\ r'(l) = 0$.
  - Protože $s'(v_i) = s(v_i) + 1 \leq s(v_{i-1}) \implies r'(v_i) \leq r(v_{i-1})$, dostaneme
$$\Delta\Phi = r'(l) + \textstyle\sum_{i \in [k]} r'(v_i) - r(v_i) \leq r'(v_1) - r(v_k) \in O(\log n) \ .$$
- **Delete:**
  - Smazání listu nebo smazání rodiče jednoho vrcholu triviální.
  - Smazání rodiče dvou vrcholů: vyměníme za minimum podstromu a pak smažeme.

Theorem:

$$\text{Amortizovaná cena splaye je } \leq 3(r'(x) - r(x)) + 1 \in O(\log n)$$

Proof:

- V jedné rotaci se mění pouze rank rotovaných vrcholů.
- **Zig:** $r'(y) \leq r'(x),\ r(y) \geq r(x)$

$$A = r'(x) - r(x) + r'(y) - r(y) + 1 \leq 2r'(x) - 2r(x) + 1$$

- **Zig-zig:** $r'(y) \leq r'(x),\ r(y) \geq r(x),\ r(z) = r'(x)$

$$r(x) + r'(z) = \log s(x) + \log s'(z) \leq 2\log(s(x) + s'(z)) - 2 \leq 2r'(x) - 2$$
$$A = r'(x) - r(x) + r'(y) - r(y) + r'(z) - r(z) + 2 \leq 3r'(x) - 3r(x) + 2$$

- **Zig-zag:** $r(y), r(w) \geq r(x)$

$$r'(w) - r'(y) = \log s'(w) - \log s'(y) \leq 2\log(s'(w) + s'(y)) - 2 \leq 2r'(x) - 2$$
$$A \leq r'(x) - r(x) + r'(y) - r(y) + r'(w) - r(w) + 2 \leq 3r'(x) - 3r(x)$$

- Nejvýš jeden zig za splay.

$$3\Big(\textstyle\sum_{i\in[k]} r_i(x) - r_{i-1}(x)\Big) + 1 = 3(r_k(x) - r_0(x)) + 1$$

Corollary:

m splayů na n vrcholech trvá $O((n+m)\log n)$

Proof:

$$\textstyle\sum_{i\in[m]} R_i = \sum_{i\in[m]} A_i + \Phi_0 - \Phi_m \ ,$$
$$\textstyle\sum_{i\in[m]} A_i \in O(m\log n) \ ,$$
$$\Phi_0 - \Phi_m \in O(n\log n) \ .$$

2.2 Haldy (regulární, binomiální).

	d -ary	Fibonacci
Insert	$O(\log_d n)$	$O(1)$
Min	$O(1)$	$O(1)$
ExtractMin	$O(d\log_d n)$	$O(\log n)$
Increase	$O(d\log_d n)$	$O(\log n)$
Decrease	$O(\log_d n)$	$O(1)$
Delete	$O(d\log_d n)$	$O(\log n)$
Build	$O(n)$	$O(n)$

d -regulární halda

Definition: ***Operace***

- **Insert:** Přidáme na konec a vybubláme.
- **Delete:** Prohodíme s posledním prvkem a zabubláme.
- **Min:** Hodnota kořene.
- **ExtractMin:** Kořen prohodíme s posledním prvkem a zabubláme.
- **Increase:** Změníme hodnotu a zabubláme.
- **Decrease:** Změníme hodnotu a vybubláme.

Binomická halda

Definition: ***Binomický strom***

B_k řádu k je kořen s podstromy $B_0, ..., B_{k-1}$.

Observation: ***Binomický strom***

1. B_k má 2^k vrcholů, výšku k a max. stupeň k .
2. B_k vytvoříme tak, že dáme B_{k-1} další B_{k-1} jako poslední dítě.
3. B_k je kostra k -hyperkostky.

Definition: ***Binomická halda***

Les $T_1, ..., T_l$ binomických stromů splňujících:

1. $r(T_i) < r(T_{i+1})$
2. Každý vrchol obsahuje jediný prvek.
3. Každý strom má haldové seřazení.

Observation: ***Binomická halda***

1. $r(T_i) \leq \log n \implies \leq 1 + \log n$ stromů
2. Přitomnost stromů podle bitů $\langle n \rangle_2$.

Definition: ***Reprezentace vrcholu***

1. Prvek a priorita
2. Rank jako počet dětí
3. Pointer na první dítě
4. Pointer na rodiče
5. Pointery na dva vedlejší sourozence (cyklicky).
6. Stromy spojuje pseudokořen

Definition: ***Operace***

- **Merge:** $O(\log n)$ Sléváme stromy od nejmenších.
  - Jediný daného řádu rovnou do výsledné haldy.
  - Dva stejného řádu sloučíme pod menší z kořenů.
  - Pro tři stejné sloučíme dva z nich a třetí rovnou do výsledné haldy.
- **Insert:** $O(\log n)$ Vytvoříme B_0 a sloučíme.
- **Delete:** $O(\log n)$ Decrease na $-\infty$ a ExtractMin.
- **Min:** $O(\log n)$ Najdeme nejmenší kořen.
- **ExtractMin:** $O(\log n)$
  - Najdeme minimální kořen.
  - Odstraníme kořen, strom se rozpadne na haldu.
  - Sloučíme haldy.
- **Increase:** $O(\log n)$ Delete a insert.
- **Decrease:** $O(\log n)$ Vybubláváme uvnitř stromu.
- **Build:** $O(n)$ amortizovaně
  - Vkládáme prvky po jednom.
  - V půlce případů $O(1)$. Každé spojení stromů odstraní jeden strom, takže umartizujeme.

Definition: ***Lazy operace***

- **Merge:** $O(1)$ Přidáme spoj.
- **Insert:** $O(1)$
- **Konsolidace:** $O(t + \log n)$
  - Připravíme $\log n$ chlívků.
  - Přidáváme stromy do chlívků. Pokud je chlívek zabraný, spojíme.
  - Každé spojení odstraní strom, proto $O(t)$ spojení.
- **ExtractMin:** $O(t + \log n)$
  1. Konsolidujeme
  2. Najdeme minimální kořen

3. Odstraníme kořen.
4. Merge s haldou.

- **Build:** $O(n)$

Definition: ***Lazy potenciál***

$$\Phi := \text{počet stromů v haldách.}$$

- **Merge:** $O(1)$
- **Insert:** $O(1)$
- **Konsolidace:** $O(\log n)$ protože $\Delta\Phi \leq \log n - t$.
- **ExtractMin:** $O(\log n)$ vyjma konsolidace.

Fibonacci halda

Definition: ***Decrease***

- Pokud porušíme vztah s rodičem, održízneme ho jako nový strom.
- Pokud rodič není kořen a je neoznačený, označíme.
- Pokud rodič není kořen a je označený, odřízneme ho a pokračujeme na jeho rodiče.
- Pokud smažeme kořen stromu, smažeme značky dětí.
- Proto každý nekořen ztratí nejvýš jedno dítě.

Observation: ***Fibonacciho čísla***

1. $F_n \in \theta(\varphi^n)$ pro $\varphi = \frac{1+\sqrt{5}}{2} \approx 1.618$
2. $F_0 + .. + F_k = F_{k+2} - 1$

Lemma:

$$\text{Podstrom řádu } k \text{ má aspoň } F_{k+1} \text{ vrcholů,}$$
$$\text{proto rank je } O(\log n) \text{ .}$$

Proof:

- Indukcí podle hloubky.

1. Listy mají rank 0 a jeden vrchol.
2. – $x_1, ..., x_n$ děti u v pořadí přidání pod něj.
   – $r_1, ..., r_n$ jejich ranky.
   – $\forall i \geq 2 : \ r_i \geq i - 2$
     * Některé ranky můžou chybět.
     * Když jsme přidávali x_i , u měl aspoň $i - 1$ dětí.
     * Spojování spojuje stejné ranky, takže x_i měl rank aspoň $i - 1$.
     * Od té doby x_i nebyl kořenem, ztratil nejvýš jedno dítě, proto má rank aspoň $i - 2$.
   – Z IP mají stromy aspoň $F_2, F_2, F_3, ..., F_k$ vrcholů $+1$ za u .

$$F_2 + F_2 + F_3 + ... + F_k + 1 \geq F_0 + ... + F_k + 1 = F_{k+2} \text{ .}$$

Observation: ***Analýza***

$$\Phi := \text{počet stromů v haldách + dvojnásobek onačených vrcholů.}$$

- **ExtractMin:** $O(\log n)$
- **Cut:** $O(1)$
  – Buď nekaskádujeme a přidáme $+3$.
  – Nebo kaskádujeme. Přidáme $+1$ za nový strom a -2 za odznačený vrchol.
- **Merge, Insert, Decrease:** $O(1)$
- **Delete:** Drahé.

2.3 Hašování, řešení kolizí, univerzální hašování, výběr hašovací funkce.

Definition: c ***-univerzální rodina***

$$\text{Rodina } \mathcal{H} : \mathcal{U} \to [m] \text{ je } c \text{ -univerzální,}$$
$$\text{pokud } \forall x, y \in \mathcal{U},\ x \neq y :\ P_{h \in \mathcal{H}}[h(x) = h(y)] \leq \tfrac{c}{m}\ .$$

Definition: $(k, c$ ***-nezávislá rodina)***

$$\text{Rodina } \mathcal{H} : \mathcal{U} \to [m] \text{ je } (k, c) \text{ -nezávislá,}$$
$$\text{pokud } \forall x_1, ..., x_k \in \mathcal{U} \text{ různá a } \forall i_1, ..., i_k \in [m] :$$
$$P_{h \in \mathcal{H}}[h(x_1) = i_1 \wedge ... \wedge h(x_k) = i_k] \leq \tfrac{c}{m^k}\ .$$

Lemma: ***K***

- $r \geq 2km$
- $\mathcal{H} : \mathcal{U} \to [r]$ je (k, c) -nezávislá

$$\mathcal{H} \mod m \text{ je } (k, 2c) \text{ -nezávislá.}$$

Proof:

- Máme nejvýš $\lceil \frac{r}{m} \rceil \leq \frac{r+m-1}{m}$ možností pro každé z $i_1, ..., i_k$.

$$\tfrac{c}{r^k}\left(\tfrac{r+m-1}{m}\right)^k = \tfrac{c}{m^k}\left(\tfrac{r+m-1}{r}\right)^k \leq \tfrac{c}{m^k}\left(1 + \tfrac{m}{r}\right)^k \leq \tfrac{ce^{\frac{mk}{r}}}{m^k} \leq \tfrac{2c}{m^k}$$

Hashovací funkce

Definition: ***Lineární hashovací funkce***

$$\mathcal{L} := \{h_{a,b} | a \in [p]_0 \backslash \{0\} \ \wedge\ b \in [p]_0\} \text{ pro } h_{a,b}(x) = (ax + b \mod p) \mod m\ .$$

Observation: ***Lineární hashovací funkce***

$$\mathcal{L} \text{ je 1 -univerzální.}$$

Proof:

- Pro $x \neq y$ a a, b jsou jednoznačně určené $r = ax + b \mod p,\ s = ay + b \mod p$.
- Pro fixní r máme $\lceil \frac{p}{m} \rceil - 1 \leq \frac{p+m-1}{m} - 1 = \frac{p-1}{m}$ špatných s . Celkem $\frac{p(p-1)}{m}$ špatných dvojic, pravděpodobnost jedné $\frac{1}{m}$.

Definition: ***Polynomiální hashovací funkce***

$$\mathcal{P}_k := \{h_t |\ t \in \mathbb{Z}_p^d\} \text{ pro } h_t(x) = \textstyle\sum_{i \in [k]_0} t_i x^i \text{ a } \mathbb{Z}_p \text{ těleso.}$$

Observation: ***Polynomiální hashovací funkce***

- $\mathcal{P}_k$ je $(k, 1)$ -nezávislá.
- Složitost $O(d)$
- $p \geq 2km \implies \mathcal{P}_k \mod m$ je $(k, 2)$ -nezávislá.

Proof:

- Mějme $x_1, ..., x_k \in \mathbb{Z}_p$ různá a $a_1, ..., a_k \in \mathbb{Z}_p$ chlívky.
- Právě jeden polynom stupně nejvýš k splňuje $\forall i \in [k] :\ h(x_i) = a_i$. Proto pravděpodobnost $\frac{1}{p^k}$.

Definition: ***Tabulační hashování***

$$\mathcal{T} : [2^{kt}] \to [2^m] :\ h(x) = \textstyle\bigotimes_{i \in [t]_0} T_i[x\langle ik : (i+1)k \rangle]$$

- Vygenerujeme t tabulek $T_0, ..., T_{t-1}$ s 2^k řádky a m sloupci.
- Pro aspoň dvě tabulky je 3 -nezávislá rodina.

Definition: ***Skalární součin***

$$\mathcal{S} = \{h_{t,b} |\ t \in \mathbb{Z}_p^d,\ b \in \mathbb{Z}_p\} \text{ pro } h_{t,b}(x) = t^T x + b$$

Observation: ***Skalární součin***

- S je $(2,1)$ -nezávislá.
- Složitost $O(d)$
- $p \geq 4m \implies \mathcal{S} \mod m$ je $(2,2)$ -nezávislá.

Proof:

- $a^T x_1 + b = y_1,\ a^T x + b = y_2,\ d := x_1 - x_2 \implies a^T d = y_1 - y_2$
- $\exists j :\ d_j \neq 0 \implies a_j = \frac{1}{d_j}(y_1 - y_2 - a_{\bar{j}}^T d_{\bar{j}})$
- Proto máme nejvýš p^{k-1} řešení z p^{k+1} funkcí.

Řešení kolizí

Definition: ***Separované řetězce***

- Tabulka obsahuje $x_1, ..., x_n$. Vybíráme h uniformně náhodně z c -univerzální rodiny.
- Neúspěšné hledání v přihrádce ověří průměrně $\frac{cn}{m}$ prvků. Úspěšné hledání x_i má složitost jako insert x_i .
- **Insert** ověří že y ještě v přihrádce není.
- **Find** prohledá přihrádku.
- **Delete** prohledá přihrádku.

Observation: ***Separované řetězce***

$$\text{Pro } m \in \Omega(n) \text{ máme } O(1) .$$

Definition: ***Kukaččí funkce***

- Tabulky T_1, T_2 a funkce $h_1 : \mathcal{U} \to [m], h_2 : \mathcal{U} \to [m]$.
- **Lookup:** x je v $T_1[h_1(x)]$ nebo v $T_2[h_2(x)]$.
- **Insert:**
  - Vložíme x do $T_1[h_1(x)]$.
  - Pokud tam byl y , opakujeme postup s $T_2[h_2(y)]$.
  - Po timeoutu přehashujeme.
- **Delete:** Smažeme prvek v jednom z polí.

Observation: ***Kukaččí funkce***

- $\epsilon > 0$
- $m \geq (2 + \epsilon)n$
- $t = \lceil 6 \log n \rceil$
- h_1, h_2 uniformně náhodně z t -nezávislého systému

$$\text{Insert s timeoutem } t \text{ má } \mathbb{E}[\text{čas}] \in O(1) .$$

Definition: ***Otevřená adresace***

- V každé buňce jeden prvek.
- Přístupová posloupnost $f : \mathcal{U} \times [m] \to [m]$
- **Insert:** Postupně zkouší buňky a uloží do první volné.
- **Find:** Postupně zkouší buňky dokud nenajde prvek nebo prázdnou.
- **Delete:** Pomníčky.

Definition: ***Lineární přidávání***

- Přidávací sekvence $h(x) + i \mod m$.
- WLOG $n \leq \frac{m}{3}$, maximalizuje počet přístupů.
- Bloky B velikosti 2^t na indexech podle stromu.

Observation: ***Lineární přidávání***

- Kritický blok je z $> \frac{2}{3}$ plný. $P[B \text{ kritický}] \leq (\frac{e}{4})^{\frac{|B|}{3}}$.
  - $x_1, ..., x_n$ vložené prvky, $X_1, ..., X_n$ indikátory $h(x_i) \in B$
  - $X = \sum X_i \implies \mathbb{E}[X] = \sum \mathbb{E}[X_i] = n\frac{|B|}{m} \leq \frac{|B|}{3}$
  - $P[X > 2\frac{|B|}{3}] < (\frac{e}{4})^{\frac{|B|}{3}}$ z Černoffa.
- R cyklický běh délky $2^{t+1} \implies$ jeden z $B_0, ..., B_3$ nejlevějších bloků velikosti 2^t v R je kritický.
- R běh s $h(x)$, $|R| \in [2^{t+2}, 2^{t+3}) \implies$ jeden z 12 bloků $|B| = 2^t$, kde devátý obsahuje $h(x)$, je kritický.
- $P[|R| \in [2^{t+2}, 2^{t+3})] \leq P[\text{nějaký kritický}] \leq 12P[\text{specifický kritický}] \leq 12(\frac{e}{4})^{\frac{|B|}{3}}$

Theorem: ***Lineární přidávání***

- $m = 2^k$
- $n \leq \frac{m}{3}$
- $h : \mathcal{U} \to [m]$ uniformně náhodná

$$\mathbb{E}[\#\text{ počet buněk navštívený při hledání } x] \in O(1)$$

Proof:

- BÚNO $n = \lfloor \frac{m}{3} \rfloor$. Běh R obsahuje $h(x)$.

$$\mathbb{E}[|R|] = \mathbb{E}[|R| \mid |R| \leq 3]\ P[|R| \leq 3]\ +\ \textstyle\sum_{t \geq 0}\ \mathbb{E}[|R| \mid |R| \in [2^{t+2}, 2^{t+3})]P[|R| \in [2^{t+2}, 2^{t+3})]$$
$$\leq 3 + \textstyle\sum_{t \geq 0} 2^{t+3} 12(\frac{e}{4})^{\frac{|B|}{3}} \leq 3 + 96 \sum_{i \geq 0} i(\frac{e}{4})^{\frac{i}{3}} \in O(1)$$

Chapter 3

Nelineární programování

3.1 Vlastnosti konvexních množin a konvexních funkcí.

Definition: ***Konvexita***

- $S \subseteq \mathbb{R}^n$ je konvexní pokud $\forall x, y \in S : \ \overline{xy} \in S$.
- $f : M \to \mathbb{R}$ je konvevní pokud ekvivalentně:

  1. $\forall x_1, x_2 \in M \ \forall \lambda \in [0,1] : \ f(\lambda x_1 + (1-\lambda)x_2) \leq \lambda f(x_1) + (1-\lambda) f(x_2)$
  2. $\{(x,z) \in \mathbb{R}^{n+1} : \ x \in M, \ z \geq f(x)\}$ je konvexní
  3. $f(x_2) - f(x_1) \geq \nabla f(x_1)^T (x_2 - x_1)$
  4. $\forall x \in M : \ \nabla^2 f(x) \succeq 0$

Observation: ***Konvexita***

- Množiny uzavřené na průnik.
- Funkce uzavřené na součet a škálování.

Theorem: ***Separační věta***

1. $C, D \subseteq \mathbb{R}^n$ konvexní, $C \cap D = \emptyset \implies$ oddělitelné.
2. Uzavřené a aspoň jedna omezená $\implies$ oddělitelné striktně.

Theorem: ***Konvexní optimalizace***

$$\min f(x) : \ x \in M \text{ pro } f, M \text{ konvexní:}$$

1. Lokální optima jsou globální.
2. Optimální množina je konvexní.
3. f ryze konvevní $\implies$ nejvýš jedno optimum.

3.2 Zobecnění konvexních funkcí.

Definition: ***Zobecněná konvexita***

$$f : M \to \mathbb{R} \text{ pro } \emptyset \neq M \text{ konvexní je:}$$

- **Kvazikonvexní:**

$$\forall \lambda \in [0,1] \ \forall x_1, x_2 \in M : \ f(\lambda x_1 + (1-\lambda)x_2) \leq \max(f(x_1), f(x_2))$$

- **Explicitně Kvazikonvexní:** kvazikonvexní a

$$\forall \lambda \in [0,1] \ \forall x_1, x_2 \in M, \ f(x_1) \neq f(x_2) : \ f(\lambda x_1 + (1-\lambda)x_2) < \max(f(x_1), f(x_2))$$

- **Pseudokonvexní:**

$$\forall x_1, x_2 \in M: \ \nabla f(x_1)^T(x_2 - x_1) \geq 0 \implies f(x_2) \geq f(x_1)$$

Theorem: ***Hierarchie***

1. Konvexní $\implies$ explicitně kvazikonvexní $\implies$ kvazikonvexní.
2. Konvexní $\implies$ pseudokonvexní $\implies$ kvazikonvexní.

Proof:

1. $f(\lambda x_1 + (1-\lambda)x_2) \leq \lambda f(x_1) + (1-\lambda) f(x_2) \leq \max(f(x_1, f(x_2))$
2. Spor a MVT.

Theorem: ***Charakterizace sublevelem***

$$f \text{ je kvazikonvexní} \iff \forall \alpha \in \mathbb{R}: \ L_\alpha = \{x |\ f(x) \leq \alpha\} \text{ je konvexní.}$$

Theorem: ***Charakterizace gradientem***

$$\text{Diferenciovatelná } f \text{ je kvazikonvexní} \iff \forall x_1, x_2: \ f(x_1) \geq f(x_2) \implies \nabla f(x_1)^T(x_2 - x_1)$$

Proof:

$\implies$ Definice gradientu.

$\impliedby$ Spor a MVT.

Theorem: ***Globální optima***

$$f \text{ explicitně kvazikonvexní} \implies \text{lokální optima jsou globální optima}$$

Proof:

- Pro spor $\exists y$ a $x = \lambda x^* + (1-\lambda)y$.
- Pak pro malé λ platí $f(x) \leq \max(f(x^*), f(y))$.

Theorem: ***Množina optim***

$$f \text{ kvazikonvexní} \implies \text{množina optim je } L_{f(x)} \text{ konvexní.}$$

3.3 Nutné a postačující podmínky optimality pro volné a vázané extrémy úloh nelineárního programování.

Theorem: ***Podmínky optimality***

1. x^* lokální minimum:
   - $f \in \mathcal{C}^1$ na $B(x^*, \epsilon) \implies \nabla f(x^*) = 0$
   - $f \in \mathcal{C}^2$ na $B(x^*, \epsilon) \implies \nabla^2 f(x^*) \succeq 0$
2. $f \in \mathcal{C}^2$ na $B(x^*, \epsilon)$, pak $\nabla f(x^*) = 0 \ \wedge \ \nabla^2 f(x^*) \succ 0 \implies x^*$ ostré lokální minimum.

Theorem: ***Explicitně kvazikonvexní podmínka optimality***

$$f(x) \text{ diferenciovatelná, explicitně kvazikonvexní,}$$
$$\text{pak } x \in M \text{ optimum pokud } \forall y \in M \backslash \{x\}: \ \nabla f(x)^T(y - x) > 0 \ .$$

Theorem: ***Pseudokonvexní podmínka optimality***

$$f \text{ pseudokonvexní a } \nabla f(x) = 0 \implies x \text{ optimum.}$$

Theorem: ***Pseudokonvexní podmínka optimality***

$$f \text{ diferenciovatelná a pseudokonvexní,}$$
$$\text{pak } x \in M \text{ je optimum } \Longleftrightarrow > \ \forall y \in M: \ \nabla f(x)^T(y-x) > 0 \ .$$

Lemma: ***Gordanova***

$$Ax < 0 \text{ řešitelná } \iff \ A^T y = 0, \ y \gneqq 0 \text{ neřešitelná.}$$

Lemma: ***Fritz John***

$$x \text{ lokální minimum } \implies \nexists d: \ \nabla f(x)^T d < 0 \ \wedge \ \forall i \in I(x): \ \nabla g_i(x)^T d < 0$$

Proof:

- $\exists d \implies 0 > \nabla f(x)^T d = \lim_{a \to 0^+} \frac{f(x-\alpha d)-f(x)}{\alpha} \implies f(x+\alpha d) > f(x)$ pro malé α .
- To samé pro omezení $\implies$ spor s optimalitou.

Theorem: ***Fritz John***

$$x \text{ lokální optimum } \implies \exists \mu, \lambda \gneqq 0:$$

1. $\mu \nabla f(x) + \lambda^T g(x) = 0$
2. $\lambda^T g(x) = 0$

Proof:

- Lemma a Gordanova věta.

Definition: ***KKT podmínky***

$$\exists \lambda \geq 0:$$

1. $\nabla f(x) + \lambda^T g(x) = 0$
2. $\lambda^T g(x) = 0$

Theorem: ***KKT LNZ***

- x lokální optimum
- $\{g_i(x)\}_{i \in I(x)}$ LNZ

Pak platí KKT.

Theorem: ***KKT Slater***

- x lokální optimum
- $f, \{g_i(x)\}_{i \in I(x)}$ konvexní
- $\exists$ Slater point

Pak platí KKT.

Proof:

- Z konvexity $\nabla g_i(x)(x^s - x) \leq g_i(x^s) - g(x) < 0$ a obdobně pro f .
- Pro $d = x^s - x$ z Gordanovy věty $\nabla g(x)^T \lambda = 0, \ (\lambda, \mu) \gneqq 0$ a z Fritze-Johna $\mu \neq 0$.

Theorem: ***Postačující KKT***

- $x \in M$ splňuje KKT
- f pseudokonvexní

- $\{g_i(x)\}_{i\in I(x)}$ kvazikonvexní

Pak x je optimum.

Proof:

- $y \in M \implies \forall j \in I(x):\ g_j(y) \leq 0 = g_j(x) \implies \nabla g_j(x)^T(y-x) \leq 0$ z kvazikonvexity.
- Aplikujeme pseudokonvexitu na

$$\nabla f(x)^T(y-x) = -\lambda^T \nabla g(x)^T(y-x) = -\textstyle\sum_{j\in I(x)} \lambda_j \nabla g_j(x)^T(y-x) \geq 0\ .$$

3.4 Kvadratické programování.

Problem: ***Kvadratické programování***

$$\min x^T Q x + q^T x:\ x \in M \text{ pro } M \text{ konvexní polyedr.}$$

Theorem: ***Těžkost***

$$\max x^T Q x:\ x \in M \text{ je NP-těžká.}$$

Proof: Vyjádříme set partition:

$$\max ||x||^2:\ \alpha^T x = 0,\ x \in [-1,1]\ .$$

3.5 Semidefinitní programování.

Problem: ***Semidefinitní programování***

- **P:** $f^* := \min c^T x:\ A_0 + \sum x_i A_i \preceq 0,\ A_i$ symetrické
- **D:** $d^* := \max \operatorname{tr}(A_0 Z):\ Z \succeq 0,\ \forall i \in [n]:\ c_i + \operatorname{tr}(A_i Z) = 0$
  - $L(x,Z) := c^T x + \operatorname{tr}\left(\left(A_0 + \sum x_i A_i\right)Z\right) = \operatorname{tr}(A_0 Z) + \sum (c_i + \operatorname{tr}(A_i Z))x_i$
  - $L(Z) = \inf_{x\in\mathbb{R}^n} L(x,Z) = \begin{cases} \operatorname{tr}(A_0 Z) & \forall i \in [n]:\ c_i + \operatorname{tr}(A_i Z) = 0 \\ -\infty \end{cases}$

Example: ***Semidefinitní programy***

- **LP:**

$$\begin{pmatrix} A_{1,*}x - b_1 & \dots & 0 \\ & \ddots & \\ 0 & \dots & A_{m,*}x - b_m \end{pmatrix} \preceq 0$$

- **Kuželové QP:**

$$||x||_2 \leq z \iff \begin{pmatrix} zI_n & 0 \\ 0 & z - \frac{1}{2}x^T x \end{pmatrix} \approx \begin{pmatrix} zI_n & x \\ x^T & z \end{pmatrix} \succeq 0$$

- **Vlastní čísla**

$$\lambda_{\max} = \min z:\ Iz \succeq A\ .$$

- **Max-cut APX**

Theorem: ***Slabá dualita***

$$Z \succeq 0 \implies d^* = L(Z) \leq f^*$$

Proof:

- Použijeme $A, B \succeq 0 \implies \operatorname{tr} AB \geq 0$. Protože $A = LL^T$ a $L^T BL \succeq 0$, máme $\operatorname{tr}(AB) = tr(L^T BL) \geq 0$.
- Pak $Z \succeq 0 \implies \forall x \in M : \operatorname{tr}\left(\left(A_0 + \sum x_i A_i\right) Z\right) \leq 0 \implies L(x, Z) \leq f(x)$ a proto

$$L(Z) = \inf_{x \in \mathbb{R}^n} L(x, Z) \leq \inf_{x \in M} L(x, Z) \leq \inf_{x \in M} f(x) = f^* \ .$$

Corollary:

Z konvexity můžeme použít Slatera $\exists x \in \mathbb{R}^n : \ A_0 + \sum x_i A_i \prec 0$.

3.6 Dualita v nelineárním programování.

Definition: ***Duální úloha***

- **P:** $f^* = \min f(x) : \ g(x) \leq 0, \ h(x) = 0, \ x \in D \neq \emptyset$
- $L(x, \lambda, \mu) = f(x) + \lambda^T g(x) + \mu^T h(x)$
- $L(\lambda, \mu) = \inf_{x \in D} L(x, \lambda, \mu)$
- **D:** $\max L(\lambda, \mu) : \ \lambda \geq 0, \ \mu \in \mathbb{R}^n$

Theorem: ***Slabá dualita***

$$d^* \leq f^*$$

Proof:

- $\forall \lambda \geq 0 \ \forall \mu \in \mathbb{R}^n \ \forall x \in M :$

$$L(x, \lambda, \mu) = f(x) + \lambda^T g(x) + \mu^T h(x) \leq f(x)$$

- Proto

$$L(x, \mu) = \inf_{x \in D} L(x, \lambda, \mu) \leq \inf_{x \in M} L(x, \lambda, \mu) \leq f^*$$

- Tedy $\forall \lambda \geq 0 \ \forall \mu \in \mathbb{R}^n : \ L(\lambda, \mu) \leq f^*$

Theorem: ***Silná dualita***

- $f, \{g_j\}_{j \in J}$ konvexní
- $\{h_i\}_{i \in I}$ lineární
- D konvexní
- $\exists$ Slater point

$$d^* = f^*$$

Proof:

- BÚNO $f^* \in \mathbb{R}, \ h(x) = Ax = 0$ a A má LNZ řádky.
- Množiny

$$A = \{(u, v, w) | \ \forall x \in D : \ u \geq g(x), \ v \geq h(x), \ w \geq f(x)\}$$
$$B = \{(0, 0, s) | \ s < f^*\}$$

  jsou konvexní a $A \cap B = \{(0, 0, f^*)\}$.
- Oddělíme s $\lambda^T u + \mu^T v + \nu w = \alpha$ pro $(\lambda, \mu, \nu) \neq 0$:

$$\forall (u, v, w) \in A : \ \lambda^T u + \mu^T v + \nu w \geq \alpha$$
$$\forall (u, v, w) \in B : \ \lambda^T u + \mu^T v + \nu w \leq \alpha \ .$$

- Pokud $\lambda < 0$, pak $(g(x^s) + \gamma,\ h(x^s),\ f(x^s)) \in A$ pro $\gamma > 0$ neomezená, spor s $\geq \alpha$. Obdobně $\nu \geq 0$.
- Pro $(0, 0, f^*) \in B$ platí $\nu w = \nu f^* \leq \alpha$ a dosazením

$$\forall (u, v, w) \in A :\ \nu w \geq \alpha \geq \nu f^* \ ,$$
$$\text{tedy } \forall x \in D :\ \lambda^T g(x) + \mu^T h(x) + \nu f(x) \geq \nu f^* \ .$$

1. $\nu > 0 \implies \forall x \in D :\ f^* \leq \frac{\lambda^T}{\nu} g(x) + \frac{\mu^T}{\nu} + f(x) = L(\nu, \frac{\lambda}{\nu}, \frac{\mu}{\nu})$ a $d^* \geq L(\frac{\lambda}{\nu},\ \frac{\mu}{\nu}) \geq f^*$.
2. $\nu = 0 \implies \forall x \in D :\ \lambda^T g(x) + \mu^T h(x) \geq 0 \implies \lambda^T g(x^s) \geq 0 \implies \lambda = 0 \ \wedge\ \mu \neq 0 \implies \forall x \in D :\ \mu^T h(x) \geq 0$. Z linearity a $\mu^T h(x^s) = 0$ máme $h(x) = 0$, ale $0 = \mu^T h(x) = \mu^T (Ax - b)$ je ve sporu s linearitou.

Theorem: ***Analýza citlivosti***

$$\text{Silná dualita} \implies f^*(u, v) \geq f^*(0, 0) - (\lambda^*)^T u - (\mu^*)^T v \ ,$$
$$\text{kde } f(u, v) = \min f(x) :\ g(x) \leq u,\ h(x) = v,\ x \in D \ .$$

Proof:

- $f^* = f^*(0, 0) = L(\lambda^*, \mu^*) = d^* = \inf_{x \in D} f(x) + (\lambda^*)^T g(x) + (\mu^*)^T h(x)$
- $M_{u,v} \subseteq D \implies \forall x \in M_{u,v} : \leq f(x) + (\lambda^*)^T g(x) + (\mu^*)^T v$
- Ale pak $\forall x \in M_{u,v} : \leq f(x) + (\lambda^*)^T + (\mu^*)^T \nu$ a $f^*(0, 0) \leq f^*(u, v) + (\lambda^*)^T u + (\mu^*)^T v$

Corollary:

$$\text{Silná dualita a } f^*(u, v) \text{ diferenciovatelná,}$$
$$\text{pak } \frac{\partial}{\partial u_i} f^*(u, v) = -(\lambda^*)_i \text{ a } \frac{\partial}{\partial v_j} f^*(u, v) = -(-\mu^*)_j \ .$$

Example: ***Efektivní úloha bez duality***

$$\min x^T Q x + q^T x :\ ||x|| \leq 1$$

- NP-těžké pro $||x||_2 \leq 1$, jinak lehké.

3.7 Metody řešení úloh na volný a vázaný extrém, včetně penalizačních a bariérových metod.

NEPODMÍNĚNÁ OPTIMALIZACE

Definition: ***Globální konvergence***

$$\text{Optimalizační metoda je } \textbf{globálně konvegrentní} \text{ pokud}$$
$$\forall x^0 \in \mathbb{R}^n :\ \liminf_{k \to \infty} ||\nabla f(x_k)|| = 0 \ .$$

SPÁDOVÉ SMĚRY

Algorithm: ***Spádové směry***

1. Vybereme spádový směr d_k aby $d_k^T \nabla f(x_k) < 0$ nebo $\cos \theta_k = -\frac{d_k^T \nabla f(x_k)}{||d_k||\ ||\nabla f(x_k)||} > 0$.
2. Vybereme α_k minimalizací jednorozměrné $\Phi(\alpha) := f(x_k + \alpha d_k)$ nebo aby $f(x_k + \alpha_k d_k) < f(x_k)$.

Algorithm: ***Největší spád***

$$d_k = -\nabla f(x_k)$$

Observation: ***Největší spád***

1. Globálně konvergentní

2. Pomalé

Algorithm: ***Newtonova metoda***

- Z Taylorova rozvoje

$$f(x_k + d) \approx m_k(d) := f(x_k) + d^T \nabla f(x_k) + \tfrac{1}{2} d^T \nabla^2 f(x_k) d$$

- Pro $\nabla^2 f(x_k) \succ 0$ hledáme d pro $\alpha = 1$ jako $\operatorname{argmin} m_k(d)$. Položením derivace rovné nule

$$d_k = -\left(\nabla^2 f(x_k)\right)^{-1} \nabla f(x_k) \ .$$

Observation: ***Newtonova metoda***

1. Potřebujeme Hessián
2. Není globálně konvergentní
3. Konverguje kvadraticky

Algorithm: ***Kwazinewtinovské metody***

$$B_k \approx \nabla^2 f(x_k)$$

TRUST REGION

Algorithm: ***Trust region***

- Aproximujeme f . Udržujeme poloměr Δ_k , ve kterém aproximaci věříme.
- Minimalizujeme aproximaci

$$\operatorname{argmin} m_k(x_k + d) : \ ||d|| \leq \Delta_k \ .$$

  Volíme $d := -B_k^{-1} \nabla f(x_k)$ pokud $||d|| \leq \Delta_k$. (A)
- Poměr reálné a modelované redukce $\rho_k := \frac{f(x_k) - f(x_k + d_k)}{m_k(0) - m_k(d_k)}$.

  a) $\rho_k \approx 1 \implies$ dobrá shoda, zvětšujeme.
  b) $\rho_k \approx 0 \implies$ špatná shoda, zmenšujeme.
  c) $\rho_k < 0 \implies f$ rostoucí, zamítáme.
  d) Jinak necháváme

0. $\overline{\Delta} > 0, \ \Delta_0 \in (0, \overline{\Delta}), \ \mu \in [0, \frac{1}{4})$
1. $\forall k \geq 0$:
2. Pokud $\rho_k < \frac{1}{4}$, pak $\Delta_{k+1} = \frac{1}{4} \Delta_k$
3. Jinak pokud $\rho_k > \frac{3}{4}$ a $||d_k|| = \Delta_k$, pak $\Delta_{k+1} = \min 2(\Delta_k, \overline{\Delta})$
4. Jinak $\Delta_{k+1} = \Delta_k$
5. Pokud $\rho_k > \mu$, pak $x_{k+1} = x_k + d_k$
6. Jinak $x_k = x_k$.

KONJUGOVANÉ GRADIENTY

Algorithm: ***Konjugované gradienty***

- Pro $A \succ 0$ řešíme $Ax = b$, tedy $\min ||Ax - b||_2^2$.

PODMÍNĚNÁ OPTIMALIZACE

AKTIVNÍ PODMÍNKY

Algorithm: ***Aktivní podmínky***

- Podobné simplexové metodě.
- Pro $K := \{j \in [m] |\ g_i(x_k) = 0\}$ řešíme pomocnou úlohu

$$\min f(x) :\ h(x) = 0,\ g_K(x) = 0\ .$$

- Nové aktivní podmínky přidáme do K .
- Lokální optimum x^* splňuje KKT podmínky

$$\nabla f(x^*) + \mu^T \nabla h(x^*) + \lambda^T \nabla g_K(x^*) = 0\ .$$

- Pokud $\lambda_j \geq 0$, přidáme j do K , v opačném případě odstraníme.

PENALIZAČNÍ METODA

Algorithm: ***Penalizační metoda***

- Řešíme $\min f(x) :\ x \in M$ pro f spojitou a $M \neq \emptyset$ uzavřenou.
- Pro **penalizační funkci** $p(x)\begin{cases} = 0 & x \in M \\ > 0 & x \notin M \end{cases}$ a $c > 0$ přeformulujeme na $\min g(x)$ pro $g(x) := f(x) + cp(x)$.
- Můžeme c zvětšovat jako $c_k \to \infty$.

Example: ***Penalizační funkce***

- $g(x) \leq 0 \ \to\ p(x) := \sum \max(0, g_j(x))^2$
- $h(x) = 0 \ \to\ p(x) := \sum h_j(x)^2$

Theorem: ***Penalizační metoda***

Pokud optimum x^k konvergují k x^* , pak x^* je optimum původní úlohy.

Proof:

- Pro $x^* \notin M \implies \exists k^*\ \forall k \geq k^* :\ x_k \notin M$, proto $g(x_k) \to \infty$!
- Jinak $\exists x' \in M :\ f(x') < f(x^*)$ a $\forall k \in \mathbb{N} :\ f(x') = g(x') < g(x^*) = f(x^*)$. Pro dost velké k pak $g(x') < g(x_k)$!

BARIÉROVÉ METODA

Algorithm: ***Bariérová metoda***

- Řešíme $\min f(x) :\ x \in M$ pro f spojitou a $M \neq \emptyset$ souvislou a uzavřenou.
- Pro **bariérovou funkci** $b(x)$ splňující $b(x) \to \infty$ pro $x \to \partial M$ a $c > 0$ přeformulujeme na $\min g(x)$ pro $g(x) = f(x) + \frac{1}{c_k} b(x)$.
- Opět $c_k \to \infty$.

Example: ***Bariérové funkce***

- $g(x) \leq 0 \ \to\ b(x) := -\sum \frac{1}{g_j(x)}$ nebo $-\sum \log(-g_j(x))$ u interior point.
- $X \succeq 0 \ \to\ -\log\det X$

Theorem: ***Bariérová metoda***

Pokud optimum x^k konvergují k x^* , pak x^* je optimum původní úlohy.

Proof: Jinak $\exists x' \in M :\ f(x') < f(x^*)$ a $\exists x'' \in \int M :\ f(x'') < f(x^*)$. Pro dost velké k pak $g(x'') < g(x^*)$ a pro dost velké k také $g(x'') < g(x_k)$.

3.8 Jednorozměrná optimalizace.

Problem: ***Line search***

$$\min f(x) \text{ pro } f : \mathbb{R} \to \mathbb{R} \text{ spojitou, případně diferenciovatelnou.}$$

Definition: ***Armijo-Golstein podmínky***

$$\text{Předpokládáme } f'(0) < 0$$

1. **Armijo:** $f(x) - f(0) \leq \epsilon_1 f'(0)x$
2. **Goldstein:** AC a $f(x) - f(0) \geq \epsilon_2 f'(0)x$ pro $\epsilon_2 < \epsilon_1$

Algorithm: ***Armijův test***

1. $\beta \in \{2, 10\},\ x > 0$
2. Dokud platí podmínka: $x := \beta x$
3. Pokud podmínka neplatí: $x := \frac{x}{\beta}$

Definition: ***Wolfeho podmínky***

$$\text{Předpokládáme } f'(0) < 0$$

1. **Slabý Wolfe:** AC a $f'(x) \geq \epsilon_2 f'(0)$
2. **Silný Wolfe:** AC a $|f'(x)| \geq \epsilon_2 |f'(0)|$

Algorithm: ***Newtonova metoda***

- Iterativně hledáme kořen f jako $x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}$. Pro minimalizaci chceme kořen derivace, tedy $x_{k+1} = x_k - \frac{f'(x_k)}{f''(x_k)}$.
- Ekvivalentně iterativně minimalizujeme aproximace druhého řádu. Chceme

  $$g(x_k) = f(x_k),\ g'(x_k) = f'(x_k),\ g''(x_k) = f''(x_k)\ .$$

  Volíme

  $$g(x) = f(x_k) + f'(x_k)(x - x_k) + \tfrac{1}{2} f''(x_k)(x - x_k)^2\ .$$

Chapter 4

Vícekriteriální a celočíselné programování

4.1 Různé přístupy k řešení úloh s více kritérii.

TYP I: BEZ DODATEČNÉ INFORMACE

GLOBÁLNÍ CÍLOVÁ FUNKCE

Definition: ***Globální cílová funkce*** Za předpokladu $F_i := \min_{x \in M} f_i(x) > 0$ řešíme

$$\min ||\left(\frac{f_1(x)-F_1}{F_1}, ..., \frac{f_s(x)-F_s}{F_s}\right)||_p : \ x \in M \ .$$

- $p = 1$ Zachovává linearitu a konvexitu.
- $p = 2$ Zachovává konvexitu.
- $p = \infty$ Maximum schováme do podmínek. Zachovává linearitu a konvexitu.

Theorem: ***Globální cílová funkce***

Pro x^* optimum úlohy platí $x^* \in \mathcal{E}$ pokud:

1. $1 \leq p < \infty$
2. $p = \infty$ a řešení je jednoznačné

Proof: Jinak $\exists x \in M : \ f(x) \lneqq f(x^*)$ a $\forall i \in [s] : \ \frac{f_i(x)-F_i}{F_i} \leq \frac{f_i(x^*)-F_i}{F_i}$ a aspoň jedna ostrá.

1. Pak $\sum_{i \in [s]} \left(\frac{f_i(x)-F_i}{F_i}\right)^p < \left(\frac{f_i(x^*)-F_i}{F_i}\right)^p$ je spor.
2. Pak $\max_{i \in [s]} \frac{f_i(x)-F_i}{F_i} \leq \max_{i \in [s]} \frac{f_i(x^*)-F_i}{F_i}$ je ve sporu s jednoznačností.

BENSONOVA METODA

Definition: ***Bensonova metoda***

$$\max e^T y : \ 0 \leq y \leq f(x^0) - f(x), \ x \in M \text{ pro } x^0 \in M \ .$$

- Zachovává linearitu a konvexitu.
- Testujeme eficientnost x^0 a hledá jiné eficientní řešení.
- Vždy přípustné pro $(x^0, 0)$.

Theorem: ***Bensonova metoda***

1. $x^0 \in \mathcal{E} \iff$ optimální hodnota je 0 .
2. (x^*, y^*) optimum $\implies x^* \in \mathcal{E}$
3. Úloha neomezená na M konvexní $\implies$

a) $\mathcal{E}^V = \emptyset$
b) $f(M)$ uzavřená $\implies \mathcal{E} = \emptyset$

Proof:

1. $\implies$ Jinak $\exists (x,y):\ e^T y > 0$, tedy $y = f(x^0) - f(x) \gneqq 0$ a $f(x^0) \gneqq f(x)$.
   $\impliedby$ Jinak $\exists x \in M:\ f(x) \lneqq f(x^0)$ a pro $y := f(x^0) - f(x)$ máme $e^T y > 0$.
2. Jinak $\exists x \in M:\ f(x) \lneqq f(x^*)$ a pro $y := f(x^0) - f(x) \gneqq f(x^0) - f(x^*) = y^* \geq 0$ máme $e^T y > e^T y^0$.
3. a) – Jinak $\exists x^*$ optimum $\max_{x \in M} \lambda^T f(x):\ x \in M$ pro $\lambda > 0$ a z neomezenosti

$$\forall \alpha > 0\ \exists x, y:\ 0 \leq y = f(x^0) - f(x),\ x \in M,\ e^T y > \alpha\ .$$

– Pro $\lambda_{\min} = \min \lambda_i > 0$ máme $\lambda^T y \geq \lambda_{\min} e^T y > \lambda_{\min} \alpha$. Dosazením $\alpha := \frac{\lambda^T (f(x^0) - f(x^*))}{\lambda_{\min}}$ dostaneme $\lambda^T (f(x^0) - f(x)) > \lambda^T (f(x^0) - f(x^*))$, tedy $\lambda^T f(x) < \lambda^T f(x^*)$.

TYP II: INFORMACE PŘEDEM

VÁŽENÝ SOUČET (SKALARIZACE)

ϵ -OMEZENÍ

Definition: ϵ ***-omezení***

- Dostaneme $\epsilon_1, ..., \epsilon_s$ maxima.
- Máme $M_k := \{x \in M |\ \forall i \neq k:\ f_i(x) \leq \epsilon_i\}$

$$P_k\ \mathbf{:}\ \min f_k(x):\ x \in M_k\ .$$

Theorem: ϵ ***-omezení***

1. x^* jednoznačné optimum $\implies x^* \in \mathcal{E}$.
2. Pro $x^* \in M:\ x^* \in \mathcal{E} \iff \exists \epsilon\ \forall k:\ x^*$ optimum P_k

Proof:

1. Jinak $\exists x \in M:\ f(x) \lneqq f(x^*)$ a $x \in M_k$.
2. $\implies$ Jinak $\exists x \in M_k:\ f_k(x) < f_k(x^*)$, tedy $x \in M$. Pro $\epsilon := f(x^*)$ máme $f_i(x) \leq e_i = f_i(x^*)$, tedy $f(x) \lneqq f(x^*)$.
   $\impliedby$ Jinak $\exists x \in M:\ f(x) \lneqq f(x^*)$, tedy $\exists k:\ f_k(x) < f_k(x^*)$ a $x \in M_k$.

LEXIKOGRAFICKÁ METODA

Algorithm: ***Lexikografická metoda***

Hledáme $\text{lexmin}_{x \in M} f(x)$ postupným řešením pro $k \in [s]$:

1. $k = 0,\ M_1 = M$
2. Pokud x^k optimum $\min_{x \in M_k} f(x)$ je jednoznačné, konec.
3. Jinak opakujeme s:
4. $k = k + 1$
5. $M_k = \{x \in M |\ \forall i < k:\ f_i(x) \leq f_i(x^i)\}$

Theorem: ***Lexikografická metoda***

Optimum je eficientní.

Proof:

- Jinak $\exists x \in M:\ f(x) \lneqq f(x^*)$ a $f(x) < f(x^*)$ lexikograficky.

TYP III: INFORMACE V PRŮBĚHU.

STEM

Algorithm: ***STEM***

1. $\forall i \in [s]$:
2. $F_i := \min_{x \in M} f_i(x)$ (předpokládáme $F_i > 0$)
3. $x^i := \operatorname{argmin}_{x \in M} f_i(x)$
4. $F_i^{\max} := \max_{j \in [s]} f_i(x^j)$
5. $\lambda_i := \frac{F_i^{\max} - F_i}{F_i ||C_{i,*}||}$
6. Najdeme optimum (x^*, y^*)

$$\min y : \ x \in M, \ \forall i \in [s] : \ \lambda_i(f_i(x) - F_i) \leq y \ .$$

7. Uživatel vrátí:
8. $I \subset [s]$ pro $f_i(x^*)$ vyhovující
9. $J \subset [s]$, kde se $f_i(x^*)$ může zhoršit o Δf_j
10. Opakujeme s úlohou $\min y : \ x \in M$ s:
11. $\forall i \notin I : \ \lambda_i(f_i(x) - F_i) \leq y$
12. $\forall i \in I \backslash J : \ f_i(x) \leq f_i(x^*)$
13. $\forall i \in J : \ f_i(x) \leq f_i(x^*) + \Delta f_i$

ALGORITMY DIALOGU

Algorithm: ***Algoritmus dialogu***

1. Uživatel zadá $\lambda^0, \lambda^1 > 0$
2. Řešíme $\min (\lambda^0)^T Cx : \ x \in M$
3. Řešíme $\min (\lambda^0 + t(\lambda^1 - \lambda^0))^T Cx : \ x \in M$ pro $t \in [0, 1]$
4. Postupně procházíme obory stability pro nová optima
5. Uživatel může změnit koncový směr

TYP IV: INFORMACE PO SKONČENÍ.

VNITŘNÍ A VNĚJŠÍ APROXIMACE

Definition: ***Vnitřní a vnější aproximace***

- A^I **vnitřní aproximace** $\forall y \in f(\mathcal{E}) \ \exists z \in A^I : \ y \lneqq z$
- A^O **vnější aproximace** $\forall y \in f(\mathcal{E}) \ \exists z \in A^0 : \ z \lneqq y$

Observation: ***Vnitřní a vnější aproximace***

$$f(\mathcal{E}) \subseteq A^0 + \mathbb{R}_+^s \cap A^I + R_-^s$$

Example: ***Odhady***

- **Jedobodový odhad:** F jako A^0
- **Konvexní odhad:** pro konvexní úlohy
  - Pro A^I vezmeme konvexní obal bodů z $\mathcal{E}$
  - Pro A^O vezmeme tečné nadroviny k bodům z $\mathcal{E}$
- **Intervalová obálka:** pomocí BB obalíme boxy.

NORMAL BOUNDARY INTERSECTION

Definition: ***CHIM***

$$\text{CHIM} := \text{conv}\{f(x^1), ..., f(x^2)\} \text{ pro } f(x^i) := \text{argmin}\, f_i(x) : \ x \in M \ .$$

Algorithm: ***Normal boundary intersection***

1. Vybereme $\binom{p+s-1}{s-1}$ bodů z CHIM rozdělením $[0,1]$ na p částí.
2. Spočteme normálu jako $v \in \text{Ker}\begin{pmatrix} f(x^2)-f(x^1) \\ \cdots \\ f(x^s)-f(x^1) \end{pmatrix}$.
3. Najdeme průměr na hranici $x_i := \text{argmax}\, \alpha : \ f(x^i) + \alpha d = f(x),\ x \in M$.
4. Nemusíme dostat eficientní řešení, lze použít Bensonovu metodu.

4.2 Funkcionál přiřazený k dané úloze vektorového programování.

Problem: ***Skalarizace***

$$\min \lambda^T f(x) : \ x \in M$$

Theorem: ***Eficientní řešení***

$$\bigcup_{\lambda \gneqq 0} M_{\text{OPT}}(\lambda) \subseteq \mathcal{E}$$

Proof:

- Jinak pro $\lambda \gneqq 0$ a $x^* \in M_{\text{OPT}}$ a $x^* \notin \mathcal{E}$ máme $\exists x : \ f(x) \lneqq f(x^*)$ a $\lambda^T f(x) < \lambda^T f(x^*)$.

Theorem: ***Vlastní eficientní řešení***

$$\bigcup_{\lambda \gneqq 0} M_{\text{OPT}}(\lambda) \subseteq \mathcal{E}^V$$

Proof:

- Jinak pro $\lambda \gneqq 0$ máme $x^* \in M_{\text{OPT}}$ a $x^* \notin \mathcal{E}^V$.
- Pro $\beta := (s-1)\max_{i,j} \frac{\lambda_i}{\lambda_j}$ musí $\exists x \in M\ \exists i \in [s] : \ f_i(x) < f_i(x^*)$ a $\forall k \neq i : \ f_i(x^*) - f_i(x) > \beta(f_k(x) - f_k(x^*))$.
- Pak $\forall k \neq i$:

$$f_i(x^*) - f_i(x) > (s-1)\tfrac{\lambda_k}{\lambda_i}(f_k(x) - f_k(x^*))$$

a proto

$$\begin{gathered} \textstyle\sum_{k\neq i} \lambda_i(f_i(x^*) - f_i(x)) > (s-1)\sum_{k\neq i} \lambda_k(f_k(x) - f_k(x^*)) \\ \lambda_i(f_i(x^*) - f_i(x)) > \textstyle\sum_{k\neq i} \lambda_j(f_k(x) - f_k(x^*)) \\ \lambda^T f(x^*) > \lambda^T f(x) \ . \end{gathered}$$

Observation: ***Inkluze***

- **Konvexní:**

$$\bigcup_{\lambda \gneqq 0} M_{\text{OPT}}(\lambda) = \mathcal{E}^V \subseteq \mathcal{E} \subseteq \bigcup_{\lambda \geq 0} M_{\text{OPT}}(\lambda) \ .$$

- **Ryze Konvexní:**

$$\bigcup_{\lambda \gneqq 0} M_{\text{OPT}}(\lambda) = \mathcal{E}^V \subseteq \mathcal{E} = \bigcup_{\lambda \geq 0} M_{\text{OPT}}(\lambda) \ .$$

- **Lineární:**

$$\bigcup_{\lambda \gneqq 0} M_{\text{OPT}}(\lambda) = \mathcal{E}^V = \mathcal{E} \subseteq \bigcup_{\lambda \geq 0} \ .$$

4.3 Pareto-optimální řešení.

Problem: ***Vícekriteriální problém***

$$\min f(x) : \ x \in M \text{ pro } M \subseteq \mathbb{R}^m \text{ a min pro všechny složky.}$$

Definition: ***Eficientní řešení***

- $\mathcal{E} = \{x \in M | \ \nexists y \in M : \ f(y) \lneqq f(x)\}$
- $\mathcal{E}^V = \{x \in M | \ \exists \beta > 0 \ \forall i \in [s] \ \forall y \in M : \ f_i(y) < f_i(x) \implies \exists k \in [s] : \ f_k(y) > f_k(x) \text{ a } f_i(x) - f_i(y) \leq \beta(f_k(y) - f_k(x))$

Observation: ***Eficientní řešení***

1. $y \in f(\mathcal{E}) \iff \mathbb{R}^s_-(y) \cap f(M) = \{y\}$
2. $f(\mathcal{E}) \subseteq \partial f(M)$

Theorem: ***Charakterizace eficientních řešení***

$$x \in \mathcal{E} \iff \forall i \in [s] : \ f_i(x) = \min f_i(y) : \ y \in M, \ \forall k \neq i : \ f_k(y) \leq f_k(x)$$

Proof:

$\implies$ Jinak $\exists i \in [s] \ \exists y \in M : \ f_k(y) \leq f_k(x) \ \wedge \ f_i(y) < f_i(x)$, spor.

$\impliedby$ Jinak $\exists y \in M : \ f_k(y) \leq f_k(x) \ \wedge \ f_i(y) < f_i(x)$ je ve sporu s optimalitou úlohy i .

4.4 Úlohy lineární a nelineární vektorové optimalizace.

Konvexní vícekriteriální programy

Problem: ***Konvexní vektorový program***

$$\min f(x) : \ x \in M \text{ pro } f, M \text{ konvexní.}$$

Theorem: ***Konvexní eficientní řešení***

$$\mathcal{E} \subseteq \bigcup_{\lambda \geq 0} M_{\mathrm{OPT}}(\lambda)$$

Proof:

- Mějme $x^* \in \mathcal{E}, \ y^* = f(x^*)$ a $K = \{y \in \mathbb{R}^s | \ \exists x \in M : \ y \geq f(x)\}$.
- $K \cap \mathbb{R}^s_-(y^*) = \{y^*\}$

  $\supseteq$ Triviální

  $\subseteq$ Jinak $\exists y^* \neq y \in K \cap R^-_s(y^*) \implies \exists x \in M : \ f(x) \leq y \lneqq y^*$ je spor s $x^* \in \mathcal{E}$.

- Oddělíme

  a) $\forall y \in K : \ \lambda^T y \geq \lambda^T y^*$

  b) $\forall y \in R^-_s(y^*) : \ \lambda^T y \leq \lambda^T y^*$.

- Z b) $\lambda \geq 0$ a $y^* - e_i \in \mathbb{R}^s_-(y^*)$. Dosadíme $\lambda^T(y^* - e_i) \leq \lambda^T y^*$ a $\lambda_i \geq 0$.
- Z a) $\forall y \in K : \ \lambda^T y \geq \lambda^T y^*$ tedy $\forall x \in M : \ \lambda^T f(x) \geq \lambda^T f(x^*)$.

Observation: ***Konvexní eficientní řešení***

$$\lambda \geq 0 \text{ a } M_{\mathrm{OPT}}(\lambda) = \{x^*\} \text{ jednoznačné optimum } \implies x^* \in \mathcal{E}$$

Proof: Jinak $\exists x \in M : \ f(x) \lneqq f(x^*) \implies \lambda^T f(x) \lambda^T f(x^*)$ je spor.

Corollary: ***Ryzí konvexita***

$$f \text{ ryze konvexní} \implies \mathcal{E} = \bigcup_{\lambda \geq 0} M_{\text{OPT}}(\lambda)$$

Theorem: ***Geoffrion***

$$\mathcal{E}^V = \bigcup_{\lambda > 0} M_{\text{OPT}}(\lambda)$$

Proof:

$\supseteq$ Už víme ze skalarizace.

$\subseteq$
- $x^* \in \mathcal{E}^V \implies \exists \beta \geq 0 \; \forall x \in M \; \forall i \in [s] : \; f_i(x) < f_i(x^*)$
- Pak $\exists k \in [s] : \; f_k(x) > f_k(x^*) \; \wedge \; f_i(x^*) - f_i(x) \leq \beta(f_k(x) - f_k(x^*))$. Neboli $f_i(x^*) + \beta f_k(x^*) \leq f_i(x) + \beta f_k(x^*)$.
- Pro fixní i mějme $F^i_j(x) := \begin{pmatrix} f_i(x) + \beta f_j(x) & i \neq j \\ f_i(x) & i = j \end{pmatrix}$ konvexní funkci.
- $\nexists x \in M : \; F^i(x) < F^i(x^*)$ jinak spor s prvním bodem.
- $y^i = F^i(x^*)$ a $K^i = \{y | \; \exists x \in M : \; F^i(x) \leq y\}$ konvexní.
- $\int K^i \cap \int \mathbb{R}^s_- = \emptyset$ jinak $\exists x \in M : \; F^i(x) \leq y \leq y^i$ pro y v průniku.
- Oddělíme
  a) $\forall y \in K^i : \; (\lambda^i)^T y \geq (\lambda^i)^T y^*$
  b) $\nexists y \in \mathbb{R}^s_-(y^i) : \; (\lambda^i)^T y \leq (\lambda^i)^T y^*$
- Znormujeme na $e^T \lambda^i = 1$ a $\lambda^i \geq 0$ z b).
- Z a) $\forall x \in M : \; (\lambda^i)^T F^i(x) \geq (\lambda^i)^T F^i(x^*)$ tj

$$\sum_{j \in [s]} \lambda^i_j f_i(x) + \sum_{j \neq i} \lambda^i_j \beta f_j(x) \geq \sum_{j \in [s]} \lambda^i_j f_i(x) + \sum_{j \neq i} \lambda^i_j \beta f_j(x^*) \; ,$$
$$\sum_{j \neq i} \lambda^i_j \beta f_j(x) \geq \sum_{j \neq i} \lambda^i_j \beta f_j(x^*) \; .$$

- Přes všechna i

$$\sum_{i \in [s]} f_i(x) + \sum_{j \neq i} \lambda^i_j \beta f_j(x) \geq \sum_{i \in [s]} f_i(x^*) + \sum_{j \neq i} \lambda_j \beta f_j(x^*)$$
$$\sum_{i \in [s]} f_i(x) + \sum_{i \neq j} \lambda^j_i \beta f_i(x) = \sum_{i \in [s]} (1 + \sum_{j \neq i} \beta \lambda^j_i) f_i(x) \geq \sum_{i \in [s]} (1 + \sum_{j \neq i} \beta \lambda^i_j) f_i(x^*)$$

- Pro $0 < \lambda^*_i := 1 + \sum_{j \neq i} \beta \lambda^j_i$ dostaneme $\forall x \in M$:

$$(\lambda^*)^T f(x) \geq (\lambda^*)^T f(x^*) \; .$$

Theorem: ***Postačující KKT podmínky***

$$x^* \in \mathcal{E}^V \implies \exists \lambda, \nu \geq 0 :$$

1. $\lambda^T \nabla f(x^*) + \nu^T \nabla g(x^*) = 0$
2. $\nu^T g(x^*) = 0$

Proof:

- $x^* \in \mathcal{E}^V \implies$ řeší $\min \lambda^T f(x) : \; g(x) \leq 0$ pro $\lambda \geq 0$.

Theorem: ***Postačující KKT podmínky***

$x^* \in M$ a splňuje KKT podmínky:

1. $\lambda \gneqq 0, \; \nu \geq 0$
2. $\lambda^T \nabla f(x^*) + \nu^T \nabla g(x^*) = 0$
3. $\nu^T g(x^*) = 0$

$$\text{Pak } x^* \in \mathcal{E}^V \; .$$

Proof:

- Podle KKT podmínek x^* řeší $\min \lambda^T f(x) : \; g(x) \leq 0$.

Lineární vícekriteriální programování

Problem: ***Lineární vektorový program***

$$\min Cx : \ Ax = b, \ x \geq 0 \ .$$

Theorem: ***Vlastnosti eficientních řešení***

$$\text{Pro } x^0 \in M \text{ a (*): } \max e^T y : \ x \in M, \ Cx + y = Cx^0, \ y \geq 0 :$$

1. $x^0 \in \mathcal{E} \iff$ úloha má optimum $(x^0, 0)$.
2. (x^*, y^*) optimum úlohy $\implies x^* \in \mathcal{E}$.
3. $\mathcal{E} = \emptyset \iff$ úloha neomezená.

Proof:

1. $\implies$ Jinak x a $y \gneqq 0$ přípustné v (*) a $Cx \lneqq Cx^0$.
   $\impliedby$ Jinak pro $Cx \lneqq Cx^0$ máme $y := Cx^0 - Cx \gneqq 0$ a $e^T y > 0$.
2. Jinak pro $Cx \lneqq Cx^* \leq Cx^0$ máme $y = Cx^0 - Cx \gneqq Cx^0 - Cx^* = y^*$, tedy $e^T y > e^T y^*$.
3. $\implies$ Omezená má optimum, které je eficientní.
   $\impliedby$
   - Jinak $x^1 \in \mathcal{E}$ a z Isermanna $\exists \lambda > 0 \ \forall x \in M : \ \lambda^T C x^1 \leq \lambda^T C x$.
   - Z neomezenosti $\exists \alpha \in \mathbb{R} \ \exists x, y : \ x \in M, \ Cx + y = Cx^0, \ e^T y > \alpha$. Pro $\lambda_{\min} := \min \lambda_i > 0$ máme spor
   $$\lambda^T (Cx^0 - Cx) = \lambda^T y > \lambda_{\min} \alpha$$
   $$\lambda^T C x < \lambda^T C x^0 - \lambda_{\min} \alpha$$

Theorem: ***Isermann***

$$\bigcup_{\lambda \gneqq S} M_{\text{OPT}}(\lambda) = \mathcal{E}$$

Proof:

$\subseteq$ Z KKT podmínek.

$\supseteq$
- Pro $x^0 \in \mathcal{E}$ máme
$$0 = \max e^T y : \ x \in M, \ y = Cx^0 - Cx \geq 0$$
- Platí $b^T u^* + (Cx^0)^T v^* = 0$ pro optimum duálu
$$\min b^T u + (Cx^0)^T v : \ A^T u + C^T v \geq 0, \ v \geq e \ .$$
- Pro $v := v^*$ je u^* optimum
$$\min b^T u : \ A^T u \geq -C^T v \ .$$
- Duál
$$-\min (v^*)^T C x : \ Ax = b, \ x \geq 0$$
má optimum x^0 , protože
$$(-C^T v^*)^T x^0 = -(x^0)^T C^T v^* = b^T u^* \ .$$

Observation: ***Hledání řešení***

1. Pokud $\mathcal{E} \neq \emptyset$, tak najdeme eficientní řešení z (*).
2. Aby skalarizace s $\lambda > 0$ byla omezená, duál $A^T y \leq C^T \lambda, \ \lambda \geq e$ musí mít řešení.

Theorem: ***Geometrie optim***

$$\mathcal{E} \text{ je sjednocení navazujících stěn } M \ .$$

Proof:

- $M_{\text{OPT}}(\lambda)$ je stěna M .
- Spojité posouvání λ chodí mezi kužely určující obory stability pro různé báze.

4.5 Metody pro získání Pareto-optimálních řešení.

4.6 Úlohy lineárního programování s podmínkami celočíselnosti, resp. s binárními proměnnými.

Problem: ***IP***

- **MILP** $\max c^T x:\ Ax \le b,\ A,b,c \in \mathbb{Q},\ C \subseteq [n],\ \forall i \in C:\ x_i \in \mathbb{Z}$
- **IP** $C = [n]$
- **bIP** $l \le x \le u$
- **BP** $x \in \{0,1\}^n$

Observation: ***MILP***

$$\textbf{Omezený IP} \to \textbf{BP}\ 0 \le x \le u \implies x_i = \sum_{k=0}^{\lceil \log u \rceil} 2^k (\langle x_i \rangle_2)_k$$

UM / TUM

Theorem: M_I

$$M_I := \operatorname{conv} M \cap \mathbb{Z}^n \text{ je konvexní polyedr.}$$

Proof:

- $x \in M \implies x = \sum \alpha_i v_i + \sum \beta_j h_j$

$$Q := \{\textstyle\sum \alpha_i v_i + \sum \beta_j h_j |\ \alpha_i \ge 0,\ e^T\alpha = 1,\ \beta_j \in [0,1]\}$$
$$C := \{\textstyle\sum \beta_j h_j |\ \beta_j \ge 0\}$$

- $M = Q + C$, chceme ukázat $M_I = Q_I + C$

  $\subseteq$ Máme $x \in M \cap \mathbb{Z}^n \implies x - \sum \lfloor \beta_j \rfloor h_j = \sum \alpha_i v_i + \sum \{\beta_j\} h_j \in Q \cap \mathbb{Z}^n$. Proto $x \in (Q_I + C)$ a $M_I = \operatorname{conv}(M \cap \mathbb{Z}^n) \subseteq (Q_I + C)$.

  $\supseteq$ $Q_I + C \subseteq M_I + C = M_I + C_I \subseteq (M + C)_I = M_I$.

Definition: ***UM / TUM***

- $A \in \mathbb{Z}^{m\times n}$ je **UM** pokud $\forall mxm$ subdeterminant je v $\{-1,0,1\}$.
- $A \in \mathbb{Z}^{m\times n}$ je **TUM** pokud $\forall$ subdeterminant je v $\{-1,0,1\}$.

Theorem: ***Hoffman / Kruskal***

1. $P := \{x|\ Ax = b,\ x \ge 0\}$ pro $A \in \mathbb{Z}^{m\times n}$ řádu m je celočíselný $\forall b \in \mathbb{Z}^m \iff A$ je UM.
2. $P := \{x|\ Ax \le b,\ x \ge 0\}$ pro $A \in \mathbb{Z}^{m\times n}$ je celočíselný $\forall b \in \mathbb{Z}^m \iff A$ je TUM.

Proof:

1. $\implies$ Mějme $x = y + A_B^{-1} e_i \ge 0$ pro B přípustnou, $y \in \mathbb{Z}^n$ dost velký a $b := A_B x = A_B y + e_i \in \mathbb{Z}^n$. Pak x doplněný nulama je vrchol, proto $A_B^{-1} e_i = x - y \in \mathbb{Z}^n$. Tedy $A_B^{-1} \in \mathbb{Z} \ \wedge\ \det(A_B)\det(A_B^{-1}) = 1 \implies \det(A_B) = 1$.

   $\impliedby$ Pro x vrchol s bází B máme $A_B x_B = b$ a $x \in \mathbb{Z}^n$ z Cramerova pravidla $x_i = \frac{\det(A_i)}{\det(A)}$.

2. Použijeme 1. na $\{x|\ Ax + Ix' = b,\ x, x' \ge 0\}$, která je UM.

Theorem: ***UM / TUM***

1. Neorientovaná I_G je TUM $\iff$ G je bipartitní.
2. Orientovaná I_G je TUM

Proof:

1. $\Longrightarrow$ BÚNO bez izolovaných vrcholů. Polyedr daný $A^T x = e,\ x \geq 0$ je neprázdný ($\frac{1}{2}e$) a omezený (suma řádků $a^T x = |V|$ pro $a > 0$) . Celočíselný vrchol je řešení dávající partity podle $x_i = 0$.
   $\Longleftarrow$ Indukcí podle velikosti podmatice.
   a) Dvě jedničky v každém sloupci dají stejné součty řádků podle partit a det $= 0$.
   b) Pokud máme někde jednu jedničku, pak Laplaceův rozvoj a IP.

Theorem: ***Cook***

IP je silně NP-těžké.

Proof:

1. Těžkost redukcí SAT. $(x \vee \neg y) \to x + (1 - y) \geq 1$.
2. Pro φ složitost zápisu řádku $(A|b)$. Pokud $M \cap \mathbb{Z}^n \neq \emptyset$, pak $\exists x^* \in \mathbb{Z}^n$ vrchol M_I se složitostí $O(\varphi n^2)$.

METODY

Algorithm: ***l-metoda***

- **Tabulka:** $\begin{bmatrix} -c & 0 \\ -I & 0 \\ A & b \end{bmatrix}$
- **Inicializace:** Pokud neplatí $R_1, ..., R_n \succ 0$, přidáme $e^T x \leq L$ a provedeme na něm pivot.
- **Iterace:**
  - Pro
  $$k = \text{argmin}\{R_{i,0} |\ R_{i,0} < 0\},\ R_l := \text{arglexmin}\{\tfrac{R_j}{|R_{k,j}|} | R_{k,j} < 0\}$$
  pivot
  $$R'_l := \tfrac{1}{-R_{k,l}} R_l, \quad R'_j := R_j - \tfrac{R_{k,j}}{R_{k,l}} R_l \ .$$
  - $R_l \succ 0\ \wedge\ R_{k,l} < 0\ \wedge\ R_{k,0} < 0\ \wedge\ R'_0 = R_0 - \frac{R_{k,0}}{R_{k,l} R_l} \implies R'_0 = R_0 \frac{R_{k,0}}{R_{k,l}} R_l \prec R_0$.
- **Terminace:**
  1. $R_{k,0} < 0\ \wedge\ R_{k,\bar{0}} \geq 0 \implies$ neřešitelné.
  2. $R_0 \geq 0 \implies$ optimum.
  3. Neomezenost s $L \implies$ neomezená.

Algorithm: ***První Gomoryho***

1. Vyřešíme relaxaci
2. Pokud řešení není celočíselné, přidáme řez a opakujeme.

$$\sum \{R_{k,j}\} x_{N_j} = \{R_{k,j}\}$$

Theorem: ***První Gomoryho***

1. $x^* \in H^-$
2. $M \cap \mathbb{Z}^n \subseteq H^+$

Proof:

Algorithm: ***Druhý Gomoryho***

1. Vyřešíme relaxaci
2. Pokud řešení není celočíselné, přidáme řez a opakujeme.

$$\sum \gamma_j x_{N_j} = \{R_{k,0}\} \text{ pro } \gamma_j := \begin{cases} \{R_{k,j}\} & N_j \in C,\ \{R_{k,0}\} \geq \{R_{k,j}\} \\ \frac{\{R_{k,0}\}}{1-\{R_{k,0}\}}(1-\{R_{k,j}\}) & N_j \in C,\ \{R_{k,0}\} < \{R_{k,j}\} \\ R_{k,j} & N_j \notin C,\ R_{k,j} \geq 0 \\ \frac{\{R_{k,0}\}}{1-\{R_{k,0}\}}(-R_{k,j}) & N_j \notin C,\ R_{k,j} < 0 \end{cases}$$

Theorem: ***Druhý Gomoryho***

1. Pro $y \leq x + b,\ x, y \geq 0,\ x \in \mathbb{R},\ y \in \mathbb{Z}$ je $y \leq \frac{1}{1-\{b\}}x + \lfloor b \rfloor$ přípustná.
2. $x^* \in H^-$
3. $M_C \subseteq H^+$

Algorithm: ***Branch and Bound***

1. Vyřešíme relaxaci
2. Nemáme optimum $\implies$ neexistuje.
3. Optimum $x \in M_C \implies$ konec.
4. Jinak zvolíme $k \in C,\ x_k \notin M_C$ a rekurzíme na:

$$M_1 := \{x \in M |\ x_k \leq \lfloor x_k^0 \rfloor\}$$
$$M_2 := \{x \in M |\ x_k \geq \lfloor x_k^0 \rfloor + 1\}$$

Observation: ***Branch and Bound***

1. **Výběr proměnné:** $k := \text{argmax}_{i:\ x_i^* \notin \mathbb{Z}} \min(d_i^-, d_i^+)$
   a) $d_i^- = \{x_i^*\},\ d_i^+ = 1 - \{x_i^*\}$
   b) $d_i^- = d_i^+ = |c_i|$
   c) $d_i^- = \{x_i^*\}|c_i|,\ d_i^+ = (1 - \{x_i^*\})|c_1|$
2. **Řezy:** Přidáváme řádek do tabulky:
   - M_1 dá $\sum R_{k,j} x_{N_j} \geq \{R_{k,0}\}$
   - M_2 dá $\sum R_{k,j} x_{N_j} \leq \{R_{k,0}\} - 1$

4.7 Nelineární optimalizační problémy s podmínkami celočíselnosti.

BENDERSOVA DEKOMPOZICE

Problem: ***Bendersova dekompozice***

$$\max c^T x + d^T y :\ Ax + By \leq b,\ x \geq 0,\ y \in Y \neq \emptyset\ .$$

- Označíme
  - $P(y)$ **:** $\max\{c^T x |\ Ax \leq b - By,\ x \geq 0\}$
  - $D(y)$ **:** $\min\{(b - By)^T u |\ A^T u \geq c,\ u \geq 0\}$
  - $U := \{u |\ A^T u \geq c,\ u \geq 0\} \neq \emptyset$ jinak M neomezená nebo prázdná. Generátory $v_1, ..., v_V,\ h_1, ..., h_H$. Předpokládáme $(b - By)^T h_j \geq 0$, jinak $D(y)$ neomezená a $P(y)$ nepřípustná.
- Přepíšeme

$$\max_{y \in Y}\{d^T y + P(y)\}$$
$$\max_{y \in Y}\{d^T y + D(y)\}$$
$$\max(d^T y + \min_{i \in [V]}(b - By)^T v_i) :\ y \in Y,\ \forall j \in [H] :\ (b - By)^T h_j \geq 0$$
$$\max z :\ y \in Y,\ \forall i \in [V] :\ z \leq d^T y + (b - By)^T v_i,\ \forall j \in [H] :\ (b - By)^T h_j \geq 0$$

- Řešíme $*P_{i,j}$

$$\max z:\ \forall i \in I:\ z \leq d^T y + (b - By)^T v_i,\ \forall j \in J:\ (b - By)^T h_j \geq 0$$

Algorithm: ***Bendersova dekompozice***

1. Vyřeš $P_{I,J}$.
   - Optimum (y^*, z^*) .
   - Neomezená (y^*, z^*) vrchol s neomezenou hranou
   - Nepřípustná $\implies$ konec.
2. Vyřeš $D(y^*)$.
   - Neomezená s h^* v v^* , přidej do I, J a od začátku.
   - Optimum v^* .
     a) Pokud $z^* \leq d^T y^* + (b - By^*)^T v^*$, pak $\forall i \in [V]:\ z^* \leq d^T y^* + (b - By^*)^T v_i \implies z^*$ hodnota optima y^* , x^* dopočítáme.
     b) Jinak přidej do I .

LAGRANGEOVA DEKOMPOZICE

Problem: ***Lagrangeova relaxace***

- $P\ \max c^T x:\ Ax \leq b,\ Dx \leq d,\ 0 \leq x \in \mathbb{Z}^n$
- $P_u\ \max c^T x + u^T(b - Ax):\ x \in Q$
- $LR\ \min P_u:\ u \geq 0$
- $Q := \{x \in \mathbb{Z}^n |\ Dx \leq d,\ x \geq 0\}$.

Theorem: ***Lagrangeova relaxace***

1. $\forall u \geq 0:\ P \leq P_u$
2. $(LR) = \max\{c^T x |\ Ax \leq b,\ x \in \operatorname{conv} Q\}$
3. $P \leq LR \leq R$
4. Q polyedr $\implies\ LR = R$

Proof:

1. Přípustná řešení v P jsou přípustná v P_u a $c^T x \leq c^T x + u^T(b - Ax)$.
2. BÚNO $Q = \operatorname{conv}\{v_1, ..., v_V\}$. Pak

$$\min u \geq 0 P_u = \min_{u \geq 0} \max_{i \in [q]} c^T v_i + u^T(b - Av_i) = \min z:\ u \geq 0,\ \forall i \in [q]:\ z \geq c^T v_i + u^T(b - Av_i)$$
$$= \max \textstyle\sum_{i \in [q]} (c^T v_i) y_i:\ e^T y = 1,\ y \geq 0,\ \sum_{i \in [q]} (b - Av_i) y_i \geq 0$$

Definition: ***Subgradient***

Subgradient v u^* pro $f : \mathbb{R}^n \to \mathbb{R}$ konvexní je $d \in \mathbb{R}^n$ splňující $\forall u \in \mathbb{R}^n:\ f(u) \geq f(u^*) + d^T(u - u^*)$.

Theorem: ***Subgradient***

$$x^* \text{ optimum } P(u^*) \implies f(u) \text{ má v } u^* \text{ subgradient } b - Ax^* .$$

Proof:

- Pro $u \geq 0$ chceme

$$\max_{i \in [q]} (b - Av_i)^T u + c^T v_i \geq (b - Ax^*)^T u^* + c^T x^* + (b - Ax^*)^T (u - u^*) = (b - Ax^*)^T u^* + c^T x^*$$

Algorithm: ***Subgradient***

1. Vyřeš $P(u^k)$
2. $u^{k+1} = \max(u^k - \alpha_k(b - Ax_k), 0)$ pro $\alpha_k \to \infty$ a $\sum \alpha_k \to \infty$.
3. $k = k + 1$

Chapter 5

Parametrické programování a intervalové metody

5.1 Parametrická skalarizace ve vícekriteriální optimalizaci a vztah s eficientními řešeními, speciální výsledky pro konvexní a lineární úlohy.

5.2 Intervalová soustava lineárních rovnic a nerovnic, různé typy řešení.

Definition: ***Intervalová matice***

- $*A = [\underline{A}, \overline{A}] = \{A \in \mathbb{R}^{m\times n} |\ \underline{A} \leq A \leq \overline{A}\}$
- $A^c := \frac{\underline{A}+\overline{A}}{2},\ A^\Delta := \frac{\overline{A}-\underline{A}}{2}$
- $A_{s,t} := A^c - D_s A^\Delta D_t$
- $b_s := b^c + D_s b^\Delta$
- $*A^S := \{A \in *A |\ A = A^T\}$

Definition: ***Množina řešení***

$$\text{Pro } *A, *b:$$

- $\Sigma^{\leq} := \{x \in \mathbb{R}^n |\ \exists A \in *A,\ \exists b \in *b:\ Ax \leq b\}$
- $\Sigma^{=} := \{x \in \mathbb{R}^n |\ \exists A \in *A,\ \exists b \in *b:\ Ax = b\}$
- $\Sigma_S^{\leq} := \{x \in \mathbb{R}^n |\ \forall A \in *A,\ \forall b \in *b:\ Ax \leq b\}$
- $\Sigma_S^{=} := \{x \in \mathbb{R}^n |\ \forall A \in *A,\ \forall b \in *b:\ Ax = b\}$

$$\text{Pro } *A^\forall, *A^\exists, *b^\forall, *b^\exists:$$

- $\Sigma_{\forall\exists} := \{x \in \mathbb{R}^n |\ \forall A^\forall \in *A^\forall\ \forall b^\forall \in *b^\forall\ \exists A^\exists \in *A^\exists\ \exists b^\exists \in *b^\exists:\ (A^\forall + A^\exists)x = b^\forall + b^\exists\}$

Theorem: ***Oettli-Prager a Gerlach***

1. $\Sigma^{=} = \{x \in \mathbb{R}^n |\ |A^c x - b^c| \leq A^\Delta |x| + b^\Delta\}$
2. $\Sigma^{\leq} = \{x \in \mathbb{R}^n |\ A^c x \leq A^\Delta |x| + \overline{b}\}$

Corollary:

- $*Ax = *b \ \approx\ *Ax \leq *b\ \wedge\ *Ax \geq *b$
- Množina řešení je konvexní polyedr v každém ortantu.

Theorem: ***Rohn***

$$\Sigma_S^{\leq} = \{x \in \mathbb{R}^n |\ x = x^+ - x^-,\ \overline{A}x^+ - \underline{A}x^- \leq \underline{b},\ x^+, x^- \geq 0\}$$

Theorem: ***Rohn***

$$*Ax = *b,\ x \geq 0 \text{ silně řešitelná } \Longleftrightarrow\ \forall s \in \{1\}^n :\ A_{s,e}x = b_s,\ x \geq 0$$

Theorem: ***Rohn***

$$\Sigma_{\forall\exists} = \{x \in \mathbb{R}^n |\ |A^c x - b^c| \leq ((A^{\exists})^{\Delta} - (A^{\forall})^{\Delta})|x| + (b^{\exists})^{\Delta} - (b^{\forall})^{\Delta}\}$$

5.3 Vlastnosti intervalových matic: regularita, vlastní čísla.

REGULARITA

Definition: ***Regularita***

$*A$ je regulární pokud $\forall A \in *A$ je regulární.

Theorem: ***Jansson***

Pro $\Sigma \neq \emptyset$:

1. $*A$ regulární $\implies$ Σ souvislá a kompaktní
2. $*A$ není regulární $\implies$ každý komponenta souvislosti neomezená

Proof:

1. $x = A^{-1}b$ spojitá funkce na kompaktu
2. – Pokud je kompoenta K omezená, máme $A^0x^0 = b^0$ s A^0 regulární a $\exists A^1$ singulární.
   – Na úsečce z A^0 do A^1 najdeme $0 < \lambda^* := \inf\{\lambda \in [0,1] :\ A(\lambda) \text{ singulární}\}$.
   – $x(\lambda)$ je spojitá funkce $\lambda \in [0, \lambda^*)$, takže $x(\lambda) \in K$.
   – Najdeme posloupnost v $[0, \lambda^*) \to \lambda^*$ a podposloupnost aby $x(\lambda_i) \to x^* \in K$. Pak
   $$A(\lambda^*)x^* = \lim_{i\to\infty} A(\lambda_i) \lim_{i\to\infty} x(\lambda_i) = \lim A(\lambda_i)x(\lambda_i) = b^0 .$$
   – Ze singularity $A(\lambda^*)$ je množina řešení $A(\lambda^*)x^* = b^0$ neomezená a v K !

Theorem: ***Postačující podmínka***

Pokud $A^c = I_n$ a $\rho(A^{\Delta})$, pak je $*A$ regulární.

Proof:

- Pro $A \in *A$ máme $|A - I_n| = |A - A_{\Delta}| \leq A_{\Delta}$, tedy $\rho(A - I_n) \leq \rho(A^{\Delta}) < 1$.

Corollary: ***Beeck***

$$\rho(|(A^c)^{-1}|A^{\Delta}) < 1 \implies *A \text{ regulární.}$$

Theorem: ***Nutná podmínka***

$$\exists i \in [n] :\ (A_{\Delta}|(A^c)^{-1}|)_{i,i} \geq 1 \implies *A \text{ není regulární.}$$

Proof:

- Vezmeme x jako i -tý sloupec (A^c) a pak $*A(A^c)^{-1} = [I_n - A^{\Delta}|(A^c)^{-1}|,\ I_n + A^{\Delta}|(A^c)^{-1}|]$.
- Pak i -tý sloupec obsahuje nulu $\Longleftrightarrow$ $(A_{\Delta}|(A^c)^{-1}|)_{i,i} \geq 1$.

Theorem: ***Těžkost***

Test regularity $*A$ je co-NP těžké i pro $A^c \geq 0,\ A^c \succ 0,\ A^c \in \mathbb{Q}^{n\times n}$ a $A^{\Delta} = ee^T$.

VLASTNÍ ČÍSLA

Definition: ***Vlastní čísla***

$$\lambda_i(*A^S) := \{\lambda_i(A)|\ A \in *A^S\}$$

Observation: ***Vlastní čísla***

1. Kompaktní interval ze spojitosti vlastních čísel.
2. Jeden nemůže být ve vnitřku jiného.

Theorem: ***Vlastní čísla***

$$\lambda_i(*A^S) \subseteq [\lambda_i(A^c) - \rho(A^\Delta),\ \lambda_i(A^c) + \rho(A^\Delta)]$$

Proof:

- Podle Weylovy věty:

$$\text{Pro } B, C \text{ symetrické: } \lambda_i(B) + \lambda_n(C) \leq \lambda_i(B + C) \leq \lambda_i(B) + \lambda_1(C)\ .$$

- Pro $A \in *A$ máme

$$\lambda_i(A) = \lambda_i(A^c + (A - A^c)) \leq \lambda_i(A^c) + \lambda_1(A - A^c) \leq \lambda_i(A^c) + \rho(A - A^c) \leq \lambda_i(A^c) + \rho(|A - A^c|) \leq \lambda_i(A^c) + \rho(A^\Delta)\ .$$

Theorem: ***Hertz***

1. $\overline{\lambda}_1(*A^S) = \max_{s\in\{1\}^n} \lambda_i(A_{-s,s})$
2. $\underline{\lambda}_n(*A^S) = \min_{s\in\{1\}^n} \lambda_n(A_{s,s})$

Proof:

- Jinak $\exists A \in *A^S :\ \lambda_1(A) > \max_{s\in\{1\}^n} \lambda_i(A_{-s,s})$, tedy $Ax = \lambda_1(A)x$ pro $x \neq 0$ a $||x||_2 = 1$.
- Pro $s := \text{sgn}(x)$ z Rayleigh-Ritz věty:

$$\lambda_1(A) = \max_{x:\ ||x||_2=1} x^T Ax\ ,$$
$$\lambda_n(A) = \min_{x:\ ||x||_2=1} x^T Ax\ .$$

  dostaneme spor

$$\lambda_1(A) = x^T Ax = \textstyle\sum_{i,j\in[n]} x_i x_j A_{i,j} \leq \sum_{i,j\in[n]} x_i x_j (A_{-s,s})_{i,j} = x^T A_{-s,s} x \leq \max_{y:\ ||y||_2=1} y^T A_{-s,s} y = \lambda_1(A_{-s,s})\ .$$

Theorem: ***Těžkost***

Následující testy jsou NP-těžké:

1. $\overline{\lambda}_1(*A^S) > 0$
2. $\overline{\lambda}_1(*A^S) \geq 0$
3. $\underline{\lambda}_n(*A^S) < 0$
4. $\underline{\lambda}_n(*A^S) \leq 0$

5.4 Deterministická globální optimalizace, horní a dolní odhady na účelovou funkci a optimální hodnotu.

Problem: ***Globální optimalizace***

$$f^* := \min f(x) : \text{pro } h, g \text{ spojité:}$$

- $\forall i \in I: \ h_i(x) = 0$
- $\forall j \in J: \ g_j(x) \leq 0$
- $x \in *x^0$

Observation: ***Těžkost***

- Řešitelnost diofantických rovnic $p(x) = 0$ je těžká.
- Řešitelnost $p(x)^2 - \sum \sin(x_i \pi)^2 = 0$ je těžká.
- Proto minimalizace levé strany je těžká.

Algorithm: ***Branch and Bound***

1. $L := \{*x^0\}, \ S = \emptyset$
2. $c^* := \infty$
3. While $L \neq \emptyset$:
4. Vybereme $*x \in L$.
5. Kontrahujeme $*x$.
6. Pro $x \in *x$ přípustné aktualizujeme c^* .
7. Pokud $\max x_i^{\Delta} \leq \epsilon$, pak $S < - * x$.
8. Jinak $L < -\{*x_i\} < - * x$.
9. Vrátíme S

Observation: ***Kontrakce***

- Zmenšujeme $*x$, aniž bychom odstranili globální minima. Můžeme dostat $*x = \emptyset$.
- Přidáme $f(x) \leq c^*$ jako podmínku.

1. Pokud $*x \subseteq M$:
   - Přidámě podmínku $\nabla f(x) = 0$.
   - Pokud $\nabla^2 f(*x)$ neobsahuje PSD matici, vyhodíme $*x$.
2. Jinak použijeme KKT podmínky a dostaneme čtvercový systém $\exists \lambda \geq 0, \mu, x$:
   a) $\nabla f(x) + \lambda^T \nabla g(x) + \mu^T \nabla h(x) = 0$
   b) $\lambda^T g(x) = 0$
   c) $x \in M$
3. Použijeme Newtonův nebo Krawczykův kontraktor.

Observation: ***Dělení boxu***

- Dělení boxu v ose k :
  a) $k := \text{argmax } x_i^{\Delta}$
  b) $k := \text{argmax } x_i^{\Delta} \left(\frac{\nabla f(x)}{\nabla x_i}(*x) \right)^{\Delta}$
  c) Lookahead

OBRAZ FUNKCE

Algorithm: ***Intervalová aritmetika***

- Přirozené intervalové rozšíření nebo
- Pro polynomy použijeme Hornerovo schéma.

Algorithm: ***Test monotónnosti***

- Spočteme obálku $*f'_{x_i}(*x)$ a testujeme, jestli překračuje nulu.

Algorithm: ***MVF***

- Z MVT $\forall a, x \in *x \; \exists x \in *x : \; f(x) = f(a) + \nabla f(x)^T (x - a)$. Intervalové rozšíření

$$f(*x) \subseteq f(a) + \nabla * f(*x)^T (*x - a) = \sum f'_{x_i}(*x_1, ..., *x_n)(*x_i - a) \; .$$

  Pro $a = x^c$ a $\nabla * f(*x)$ isotonní v inkluzi je MVT isotonní v inkluzi.

- Těsnější obálku dostaneme po složkách

$$f(x) = f(\{x_i\}_{i\in[n-1]}, a_n) + f'_{x_n}(\{x_i\}_{i\in[n-1]}, c_n)(x_n - a_n)$$
$$=$$
$$f(\{x_i\}_{i\in[n-2]}, \{a_i\}_{i\in[n]\setminus[n-2]}) + f'_{x_{n-1}}(\{x_i\}_{i\in[n-2]}, c_{n-1}, a_n)(x_{n-1} - a_{n-1}) + f'_{x_n}(\{x_i\}_{i\in[n-1]}, c_n)(x_n - a_n)$$
$$...$$
$$= f(x) + \textstyle\sum_{k\in[n]} f'_{x_k}(\{x_i\}_{k-1}, c_k, \{a_i\}_{[n]\setminus[k]})(x_k - a_k)$$

  proto

$$f(*x) \subseteq f(a) + \textstyle\sum_{k\in[n]} f'_{x_k}(\{*x_i\}_{k-1}, c_k, \{a_i\}_{[n]\setminus[k]})(*x_k - a_k)$$

Algorithm: ***Slopes***

- Protože $f(x) = f(a) + \frac{f(x)-f(x)}{x-a}(x - a)$, definujeme

$$S_f(x, a) := \begin{cases} \frac{f(x)-f(a)}{x-a} & x \neq a \\ f'(x) & x = a \end{cases} .$$

- Intervalové rozšíření

$$f(*x) \subseteq f(a) + S_f(*x, a)(*x - a) \; .$$

- Můžeme aplikovat aritmetická pravidla pro výpočet $S_f(x, a)$

OBÁLKY NELINEÁRNÍCH SYSTÉMŮ

Algorithm: ***Intervalová linearizace***

- Z MVT $\forall a, x \in *x \; \exists c \in *x \; \forall x \in *x : \; f(x) = f(a) + \nabla f(x)^T (x - a)$. Intervalové rozšíření $\forall x \in x$

$$f(x) \subseteq f(a) + \nabla f(*x)^T (x - a) \; ,$$

  takže obalujeme lineární intervalovou funkcí. Pro $x^* \in *x$ kořen $0 \in f(x^0) + \nabla f(*x)^T (x^* - a)$

$$0 \in f(a) + \nabla f(*x)^T (x^* - a)$$

Algorithm: ***Newtonův kontraktor***

$$N(*x, x^0) := \square\{x \in \mathbb{R}^n | \; \exists A \in \nabla f(*x) : \; A(x - x^0) = -f(x^0)\}$$

- Najdeme obálku $*y$ pro $\nabla f(*x)y = -f(x^0)$ a vrátíme $N(*x, x^0) = x^0 + *y$.

Algorithm: ***Krawczykův kontraktor***

$$K(*x, x^0, C) := x^0 - Cf(x^0) + (I_n - C\nabla f(*x)(*x - x^0))$$

- Často $C \approx \nabla f(x^0)^{-1}$.

OBÁLKY PODMÍNEK

Algorithm: ***Branch and Prune***

Algorithm: ***Contraint propagation***

HORNÍ ODHADY

- **Bez rovnic:** Libovolné řešení.
- **S rovnicemi:** Zafixujeme $n - |I|$ proměnných např podle nejmenšího gradientu. Na čtvercový systém $h(x) = 0$ aplikujeme intervalovou Newtonovou metodu.

DOLNÍ ODHADY

Algorithm: ***Konvexní underestimator pro bilineární funkce***

- Pro $y \in *y,\ z \in *z$ máme $yz \geq \max\{\underline{y}z + y\underline{z} - \underline{y}\underline{z},\ \overline{y}z + y\overline{z} - \overline{y}\overline{z}\}$
- Můžeme použít pro $xyz = xy'$.

Algorithm: ***Konvexní underestimator pro obecné funkce***

- Pro $f(x) \in \mathcal{C}^2$ a parametry α definujeme $g(x) := f(x) + (x - \underline{x})^T D_\alpha (x - \overline{x})$.
- Hledáme α aby $\nabla^2 f(*x) + 2D_\alpha \succeq 0$ nebo jednodušeji β aby $\nabla^2 f(*x) + 2\beta I_n \succeq 0$.
- Položíme $\beta := -\frac{1}{2}\underline{\lambda}_n(\nabla^2 f(*x))$.