Základy spojité optimalizace

2. ledna 2013

Obsah

1 Přehled

1.1 Obecná úloha

Obecnou úlohou matematické optimalizace rozumíme úlohu maximum či minimum funkce f na množině M, kde f je obecná funkce. f se nazývá ***účelová*** nebo ***cílová***. M se nazývá ***množinou přípustných řešení***. Každý prvek $x \in M$ se nazývá ***přípustným řešením***. Takové $x_0 \in M; \forall x \in M; f(x_0) \geq f(x)$ se nazývá ***optimálním řešením***.

1.2 Dělení úloh

Dle spojitosti:

- **Diskrétní** – řeší se v kombinatorice, mohou mít konečně mnoho možností.
- **Spojité** – funkce většinou reálné proměnné.

Dle determinističnosti:

- **Deterministické**
- **Stochastické** – obsahují nějaké náhodné proměnné. Řeší statistika.

Dle volnosti:

- **S volným extrémem** – množina M není definovaná, hledá se na celém prostoru. Hledá se jinak, než u vázaných.
- **S vázaným extrémem** – omezené množinou M.
  - Lineární.
  - Nelineární – konvexní (často kvadratické), zobecněné konvexní a speciální nekonvexní.
  - Dynamické – každé podřešení optimálního řešení je také optimální. Rozhodovací proces, hledá posloupnost. Diskrétní.
  - Optimalizační procesy – Totéž, ale spojité.
  - Teorie her – někdo je protivník.
  - Semiinfinitní programování.

  Lze udělat nadstavby:

  - Celočíselné programování – některé parametry jsou diskrétní.
  - Parametrické programování

– Vícekriteriální

Pozorování 1 •

$$\max f(x) = \min -f(x)$$

Tedy, mohu se zabývat pouze jedním z tohoto.

- *Někdy je potřeba operovat se supremem či infimem.*

1.3 Volný extrémem

Hledám $\min\{f(x); x \in \mathbb{R}^n\}$.

Používají se metody newtenova, kvazinewtenova. Využívá se v penalizačních a bariérových metodách.

Penalizační je tak, že se jde „zvenčí" množiny M, penalizuji za to, že jsem venku, takže časem dojdu dovnitř. Bariérová metoda říká, že začnu někde uvnitř a nesmím M opustit.

1.4 Lineární programování

Hledám minimum:

$$\min_M \left\{C^T x, C \in \mathbb{R}^n\right\}, M = \left\{x \in \mathbb{R}^n; Ax \leq (\geq, =)b, A \in \mathbb{R}^{m+n}; b \in \mathbb{R}^m\right\}$$

Úlohou Lineárního programování v rovnicovém tvaru rozumíme úlohu:

$$\min_M \left\{C^T x\right\} \quad , \quad M = \{x \in \mathbb{R}^n; Ax = b, x \geq 0\}, A \in \mathbb{R}^{m+n},$$
$$h(A) = m, 1 \in m < n, b \geq 0, C \in \mathbb{R}^n$$

Úlohou Lineárního programování v normálním tvaru rozumíme:

$$\min_M \left\{C^T x\right\}, M = \{x \in \mathbb{R}^n; Ax \leq b, b \geq 0\}, A \in \mathbb{R}^{m+n}, b \in \mathbb{R}^m, C \in \mathbb{R}n$$

1.4.1 Simplexová metoda

Vychází z toho, že nějaké optimální řešení, pokud existuje, tak je alespoň v jednom z vrcholů mnohostěnu. Začnu v některém vrcholu, posouvám se na ty lepší a lepší, až ho najdu.

1.4.2 Duální simplexová metoda

Řeší dvě duální úlohy zároveň. Vezmu bázická (nemusí být přípustná) řešení, ale musí dávat stejnou funkční hodnotu. Měním tak dlouho, než dostanu přípustná řešení, protože jsou stejná, musí to být optimum.

Věta 1 *Existuje-li řešení, je ho dosaženo alespoň v jednom vrcholu konvexního polyedru M.*

1.5 Konvexní programování

Hledám $\min_M f(x); f(x)$ je konvexní funkce:

$$M = \{x \in \mathbb{R}^n; g_j(x) \leq 0; j \in 1 \dots n\}$$

g_j jsou podmínky, je jich konečně mnoho. Zajišťují konvexnost množiny.

Věta 2 *Každé lokální minimum konvexní funkce je absolutním minimem.*

Zobecněné, kvazikonvexní, explicitně kvazikonvexní a pseudokonvexní funkce jsou takové, že platí stejná (nebo podobná) věta.

Používají se gradientní metody (též známé jako metody přípustných směrů) – jdu „z kopce“, dokud je to dolů. Vždy najdu směr, kam se vydám a jak daleko.

Dále jsou metody sečných nadrovin – spočítám funkční hodnotu, vždy odseknu nadrovinou něco, kde se optimum nemůže nacházet.

1.6 Celočíselné programování

Nemá oddělenou teorii, staví nad lineárním a nelineárním programováním.

Používají se sečné nadroviny, zapomenu na celočíselnost, když to nevyšlo, tak ořížnu tak, aby mi zbyly celočíselné, ale ne toto.

Metoda větvení a mezí – simplexovou metodou, když najdu něco neceločíselného, vynechám čísla mezi, rozdělím na dvě množiny.

1.7 Parametrické programování

Výsledná funkce závisí na nějakém parametru.

Rozdělí se na obory stability (kdy se výsledek x nemění v závislosti na parametru).

1.8 Vícekriteriální optimalizace

$\max_M f(x)$, kde f je vektorová funkce.

$x_0 \in M$ je ***eficientní***, pokud $\nexists x \in M$, který by byl alespoň tak dobrý ve všech složkách a alespoň v jedné je lepší. Nemusí být všechny.

Přístup kompromisního řešení ohodnotí f pomocí nějaké skalární funkce a řeší se ta obvyklým způsobem.

1.9 Dynamické programování

Předmětem zkoumání jsou diskrétní rozhodovací procesy.

$$\max_{\mathcal{P}_n} f_N(x_1, x_2, \ldots, x_n)$$

Několikrát za sebou se rozhoduji. Hledám posloupnost prvků. Mám dva druhy proměnných – stavové (popisují aktuální situaci) a rozhodovací (popisují rozhodnutí).

Věta 3 (Bellmanův princip optimality) *Každá podstrategie optimální strategie je také optimální*

1.10 Optimalizační procesy

Spojité rozhodovací procesy. Nerozhoduje se v daných bodech, ale stále.

1.11 Teorie her

Je ještě protihráč. Např. maticová hra dvou hráčů s nulovým součtem – co vyhraje jeden, prohraje ten druhý. Vždy existuje pro oba hráče optimální strategie, přímá souvislost s principem duality v lineárním programováním.

1.12 Semiinfinitní programování

$$\max_M C^t x, M \in \{\mathbb{R}^n | a(t) \leq b(t)\}, t \in T \subseteq \mathbb{R}^m, a(t) \in \mathbb{R}^n, b(t) \leq \mathbb{R}, C \in \mathbb{R}^n$$

T je obvykle kompaktní a konvexní.

Problém je, že je zde nekonečně mnoho omezení.

2 Lineární programování

Uvažujeme úlohu v rovnicovém tvaru.

$$\min_{M} C^T x, M = \{x \in \mathbb{R}^n; Ax = b, x \geq 0\}, C \in \mathbb{R}^n, b \in \mathbb{R}^m, A \in \mathbb{R}^{m\times n}, h(A) = m, 1 \leq m \leq n$$

Tedy $\exists$ konečný počet (alespoň 1) $A_B \in \mathbb{R}^{m\times m}$ regulární. Vezmeme libovolnou A_B. Potom můžeme přepsat úlohu:

$$A = (A_B, A_N)$$

Stejným způsobem rozdělíme $x = (x_B, x_N), C = (C_B, C_N)$.

Protože $A_B x_B + A_N x_N = b \Rightarrow x_B = A_B^{-1} b - A_B^{-1} A_N x_N = d_0 - D x_N, C_B^T x_X + C_N^T x_N \equiv C_B^T (d_0 - D x_N) + C_N^T x_N = C_B^T d_0 + (C_N - C_B^T D) x_N = \equiv c_0 + (C_N - Z_N) x_N$.

Známe-li libovolné bázické řešení B, dá se přepsat na tvar:

$$\min_{M} \left\{C_0 + (C_N - Z_N)^T, M = \{(x_B, x_N); x_B = d_0 - D x_n \geq o, x_N \geq 0\}\right\}$$

V simplexové metodě je vždy $A_B = I_n$ jednotková matice. Buď máme štěstí a je tam a nebo je třeba ji vyrobit.

Výchozí tabulka

1	0	0	a	a	a	b
0	1	0	a	a	a	b
0	0	1	a	a	a	b
0	0	0	x	x	x	$-c_0$

a je matice A_n, x jsou $c_N - Z_n$.

Věta 4 *Platí-li $C_n - Z_n \geq 0$ v každé složce, potom příslušné bázické řešení $x = (x_B, x_N)$, $x_B = d_0, x_N = 0$ je optimální.*

Věta 5 *Existuje-li index $l \in N$ tak, že příslušné $C_n - Z_n \leq 0$ a $d_l \leq 0$ v každé složce, potom neexistuje řešení úlohy.*

Věta 6 *Nejsou-li splněny předpoklady 4 ani 5, potom označíme-li $C_S - Z_S = \min C_j - {z_j}_{j\in N}$*

$\min_{d_s > 0} \left\{\frac{d_{i0}}{d_{i-s}}\right\} \equiv \frac{d_{r0}}{d_{rs}}$, *dostaneme po transformaci tabulky s pivotem d_{rs} nové přípustné řešení $(x'_B, x'_N), C^T x' \leq C^T x$.*

Mám konečný počet bází, ale při transformaci může zůstat stejná hodnota cílové funkce, může vzniknout degenerace a cyklus.

2.1 Případ degenerace a cyklu

Přípustné bázické řešení (x_B, x_N) nazveme ***degenerovaným***, jestliže existuje alespoň jeden index $p \in B$ tak, že $x_p = 0$.

Říkáme, že při simplexové metodě dochází k degeneraci, jestliže alespoň jedno přípustné bázické řešení je degenerované. Podle transformačního vzorce platí $(-c_0)' = -c_0 - \frac{(c_s - z_s) \cdot d_{r0}}{d_{rs}}$. Pokud d_{r0}, pak se nezmění funkční hodnota. Může se to zacyklit.

Degenerace nemusí vždy nutně znamenat cyklus.

2.1.1 Blandovo pravidlo

Definujeme-li s jako $s \equiv \min \{l \in N; C_l - z_l < 0\}$ a $r \equiv \min \left\{p \in B; \min \left\{\frac{d_{i0}}{d_{is}}\right\} = \frac{d_{p0}}{d_{ps}}\right\}$. Potom při aplikaci simplexové metody nedojde k cyklu.

2.1.2 ϵ-modifikovaná úloha

Degenerace může vzniknout buď zadáním (některé $b_l = 0$) nebo nejednoznačností výběru klíčového řádku. Platí-li, že $\min_{d_{is}>0} \left\{\frac{d_{i0}}{d_{is}} = \frac{d_{r0}}{d_{rs}} = \frac{d_{p0}}{d_{ps}}\right\}$ a zvolím-li d_{rs} za pivota.

Změním množinu řešení tak, že $M_\epsilon = \left\{x \in \mathbb{R}^n; Ax = b + A\epsilon, x \geq 0, \epsilon = \left(\epsilon, \epsilon^2, \epsilon^3, \ldots, \epsilon^m\right)\right\}$ za předpokladu, že $A = (EA_N)$. Tím zajistím, že degenerace nevznikne v zadání ani v průběhu.

V praxi se to dělá tak, že když nemám jednoznačně určen klíčový řádek, tvoříme minima z podílů prvků prvního sloupce tabulky k prvkům klíčového sloupce, které připadají v úvahu na pivota. Pokud ještě něco zbude, přejdu k dalšímu sloupci a tak dále.

2.2 Množina všech optimálních řešení

Množinu $M^{opt} = \left\{x_i \in M; \min_M C^T x = C^T x_i\right\}$ nazveme množinou všech optimálních řešení.

Poznámka 1 *Pokud známe jedno optimální řešení x_0 s hodnotou cílové funkce $C^T x_0 = C_0$, potom můžu zapsat množinu $M^{opt} = \left\{x \in M; C^T x = C_0\right\}$.*

Věta 7 *Jsou-li x_1, x_2 dvě různá optimální řešení úlohy lineárního programování, potom každý bod úsečky (x_1, x_2) je také optimálním řešením.*

Důkaz:
$C^T x_1 = C^T x_2 = C_0$, platí $\forall x \in (x_1, x_2)$:

$$\begin{aligned} C^T x &= C^t(\lambda x_1 + (1-\lambda)x_2) \\ &= \lambda C^T x_1 + (1-\lambda)C^T x_2 \\ &= (\lambda + 1 - \lambda)C_0 = C_0 \end{aligned}$$

☻

Tvrzení 1 *M^{opt} je rovna uzávěru stěny M určité dimenze.*

Nechť jsme získali optimální řešení x_0 simplexovou metodou a poslední tabulka je tvaru:

Věta 8 *$M^{opt} = \{x \in M; x_j = 0; j \in J\}$. J jsou indexy, kde dole vyjde číslo > 0.*

Důkaz:
Rozepsáním přes sumu.

☻

Důsledek 1 *Pokud všechna $C_j - z_j > 0$, potom je x_0 jediným optimálním řešením.*

Poznámka 2 *Pokud v poslední (optimální) tabulce máme nějaké $C_j - z_j = 0, j \in N$, potom zkoušíme zvolit příslušný sloupec za klíčový.*

- *Najdeme-li pivota (existuje ve sloupci alespoň jedna kladná hodnota), označme ho d_{rj}, potom nové přípustné bázické řešení má hodnotu cílové funkce $-c_0' = -c_0 - \frac{(c_j - z_j) \cdot d_{r0}}{d_{rj}}$, tedy hodnota optimální funkce zůstala zachována, dostali jsme tedy nové optimální řešení.*

  Pozor, kdyby nastala degenerace, můžu zůstat ve stejném vrcholu.

- *Neexistuje-li pivot ($\forall d_{ij} \leq 0$). Potom neomezená hrana*

  $$\{x \in \mathbb{R}^n; x = x_0 + td_j; t \geq 0\}$$

  představuje optimální řešení.

- *Konvexní kombinací získaných dalších optimálních řešení najdeme část množiny M^{opt}.*

2.3 Revidovaná simplexová metoda

Chceme se zbavit jednotkové matice v tabulce. Při každé transformaci vznikne jeden nejednotkový sloupec, ale jeden se zase vyrobí. Tyto lze ignorovat. Budu si pamatovat, který sloupec je který „původní“ index – které proměnné odpovídá.

Nejprve se bude transformovat klíčový sloupec. Na místě pivota bude jeho převrácená hodnota. Ostatní prvky sloupce dělíme $-p$ (p je pivot).

Ostatní prvky tabulky zpracujeme jako u obyčejné simplexové metody. Je třeba sledovat, která proměnná to je.

2.4 Výpočetní složitost simplexové metody

Jeden krok se spočítá jednoduše. Tedy mě zajímá počet kroků.

- Lze odhadnout $\binom{n}{m}$.
- Hledáme největší úbytek hodnoty cílové funkce. Při transformaci tabulky v libovolném kroku dostaneme nové

  $$-C_0' = -C_0 - \frac{(c_s - r_s) \cdot d_{r0}}{d_{rs}}$$

  Za klíčový sloupec by měl být brán takový, pro který platí $\min_{c_j - z_j < 0} \left\{ \frac{(c_j - z_j) \cdot d_{r(j)0}}{d_{r(j)j}} \right\}$. Pomalé na zjišťování.
- Můžu použít $\min_{c_j - z_j < 0}(c_j - z_j)$. Reálně to u praktických problémů počet kroků nezávisí na počtu proměnný a pohybuje se mezi $2m$ až $3m$.

Odhady říkají, že $P(n, m) \leq m^{\frac{1}{n-1}} \cdot (m+1)^n \cdot \frac{2}{5} \cdot \pi \cdot (1 + \frac{e\pi}{2})$. V praxi bývá mnohem lepší.

2.5 Teoreticky rychlejší metody

Pokusy vytvořit polynomiální algoritmy, prakticky jsou pomalé.

2.5.1 Chačiova elipsová mateda

Sestrojí se elipsoid obsahující množinu přípustných řešení, vezme se polovina, kde M leží. Pak se sestrojí další elipsoid uvnitř této poloviny. Dělám to tak dlouho, až najdu střed elipsoidu uvnitř M.

2.5.2 Karmankarova

Vytvořil se simplex, uvnitř se opsala nadkoule M, cílová funkce se promítla na nadkouli, opět se sestrojil simplex, etc.

2.6 Duální simplexová metoda

2.6.1 Princip duality

Máme primární úlohu v základním tvaru:

$$\max_{M_1} C^T x, M_1 = \{x \in \mathbb{R}^n; Ax \le b; x \ge 0\}, C \in \mathbb{R}n, A \in \mathbb{R}^{m \times n}, b \in \mathbb{R}^m$$

Poté úloha:

$$\min_{M_2} b^T y, M_2 = \left\{y \in \mathbb{R}^m; A^T y \ge C, y \ge 0\right\}$$

je této ekvivalentní.

Lemma 1 *Jsou-li $M_1, M_2 \neq \emptyset$, potom $\forall x \in M_1, y \in M_2; c^T x \le b^T y$.*

Důkaz:
Nějaké x a y existují.

☻

Věta 9 (Princip duality) *Tyto úlohy mají stejná optimální řešení.*

Důsledek 2 *Rovnost funkčních hodnot platí právě pro optimální řešení.*