Základy spojité optimalizace

Ladislav Láska

2. března 2010

Obsah

1 Úvod

1.1 Úloha, cílová funkce, množina řešení

Definice Úloha matematického programování (optimalizace) rozumíme úlohu

$$\min_{x \in M} f(x)$$

kde $f : \mathbb{R}^n \to \mathbb{R}$.

Definice Funkci $f(x)$ nazýváme **cílovou**, účelovou, kriteriální, objektivní funkcí.

Definice Množinu M nazýváme množinou přípustných řešení. Prvek $x \in M$ nazýváme přípustným řešením optimalizační úlohy. Prvek $x_0 \in M$ nazveme optimálním řešením.

1.2 Dělení na jednotlivé disciplíny optimalizace

1. Volný extrém - $\min_M f(x)$
2. Vázaný extrém - $\min_M f(x), M \subset \mathbb{R}^n$
   (a) Lineární programování:
   $\min_M cx$, $M = \{x | A_x\{=, \leq, \geq\}b\}$, kde $c \in \mathbb{R}^n$, $A \in \mathbb{R}^{m \times n}$, $b \in b^m$
   (b) Nelineární programování:
   $\min_M f(x), M = \{x | g_j(x) < 0 (j = 1, ..., m)\}, f, g_j : \mathbb{R}^n \to \mathbb{R}$
      i. Konvexní a zobecněné konvexní programování - f, g_j konvexní, dále pak kvadratické a hyperbolické programování
      ii. Nekonvexní (speciální typy)
   (c) Celočíselné programování:
   Lineární/nelineární programování, navíc podmínky pro celočíselnost $Ax * b$, aby $x \in \mathbb{N}$.
   (d) Parametrické programování:
   Lineární/nelineární programování, navíc parametr $\min_{M(U)} c(\lambda)^T x$, $c(x) = c + C\lambda$, $M(U) = \{x | A(U)x * b(U)\}$
   (e) Vícekriteriální (vektorové) programování:
   $\min_M f(x)$, $f(x) = \{f_1(x), .., f_s(x)\}$
   (f) Dynamické programování - hledání optimální strategie
   (g) Spojité programování (optimalizační procesy)
   (h) Teorie her - optimální strategie dvou hráčů
   (i) Semiinfinitní programování - nekonečně mnoho podmínek

1.3 Motivační úloha

1.3.1 Lineární programování

$V_1, ..., V_n$ - výrobci vyrábějící výrobek V v množstvích $a_i > 0$
$S_1, ..., S_k$ - spotřebitelé požadující výrobek V v množstvích $b_j > 0$.
Známe cenu za dopravu jednotky výrobku V z V_i do S_j - $c_{i,j} \geq 0$.

Předpoklad: ceny za dopravu rostou lineárně.

Cíl: minimalizovat celkové náklady na dopravu.

Hledáme: množství $x_{i,j} \geq 0$ - kolik výrobce V_i dodá S_j.

Cílová funkce

$$f(x) = \sum_i \sum_j c_{i,j} x_{i,j} \tag{1}$$

na množině řešení

$$M = \{x_{i,j} | \sum_{i=1}^{m} x_{i,j} = b_j \forall j, \sum_{j=1}^{m} x_{i,j} = a_i \forall i, \sum_i a_i = \sum b_j, x_{i,j} \geq 0 \forall i \forall j\} \tag{2}$$

Všechno je lineární - úloha lineárního programování.

1.3.2 Celočíslené programování

Pokud nelze položky libovolně dělit (například lidi), můžeme přidat celočíselnou podmínku do množiny řešení:

$$x_{i,j} \mathbb{N}_0 \forall i \forall j \tag{3}$$

1.3.3 Nelineární programování

Zrušíme předpoklad lineárního růstu cen (tedy cena závisí na množství)

$$f(x) = \sum_i \sum_j c_{i,j}(x_{i,j}) x_{i,j} \tag{4}$$

$c_{i,j} = C_{i,j} + c_{i,j} x_{i,j}$

1.3.4 Parametrické programování

Produkce není pevná a závisí na parametru: $a_i = \lambda a'_i$. Ostatní vztahy můžou zůstat třeba jako u lineárního programování.

1.3.5 Vícekriteriální programování

Minimalizovat ceny za dopravu, maximalizovat zisky Z.

$$\min\{f(x), g(x)\} \tag{5}$$
$$f(x) = \sum_i \sum_j c_{i,j} x_{i,j} \tag{6}$$
$$g(x) = -Z(x_{i,j}) \tag{7}$$

1.3.6 Dynamické programování

Ceny závisí na rozhodnutí.

2 Volný extrém

Hledání $\min f(x)$.

2.1 Postačující podmínka

Jestliže má funkce $f(x)$ v $x_0 \in \mathbb{R}^n$ spojisté 2. parciální derivace, $\nabla f(x_0) = 0$, $\nabla^2 f(x_0)$ (Hessova matice) je pozitivně definitní, potom má reálná funkce $f(x)$ v x_0 ostré lokální minimum.

2.2 Penalizační metody

Máme úlohu

$$\min_M f(x), \quad M = \{x | g_j(x) \leq 0, j = 1, ..., n\} \tag{1}$$

Hledáme penalizační funkci $p(x)$ vyjadřující pokutu za to, že pracujeme s $x \in M$. $p(x)$ je vytvořená z podmínek g_j a řešíme posloupnost úloh na volný extrém:

$$\min\{f(x) + \alpha_k p(x)\} \tag{2}$$

Po p(x) požadujeme:

$$p(x) > 0 : \quad x \notin M \tag{3}$$
$$p(x) = 0 : \quad x \in M \tag{4}$$
$$p(x) \quad \text{spojitá} \tag{5}$$

Posloupnost takovýchto úloh za jistých okolností konverguje k optimálnímu řešení.

Příklad

$$\sum_{j=1}^{n}(g_j^+(x))^2, \quad g_j^+(x) = \max\{g_j(x), 0\} \tag{6}$$

2.3 Bariérové metody

Hledáme bariérovou funkce $b(x)$, která nám zabrání vystoupit z množiny M. Řešíme tedy posloupnost úloh:

$$\min\{f(x) + \frac{1}{\beta_k}b(x)\}, \quad \beta_k \to \infty \tag{1}$$

A po funkce b(x) požadujeme:

$$b(x) \leq 0 : \quad x \in M \lim_{x \to \partial M} b(x) = \infty \tag{2}$$

kde ∂M je hranice množiny.

Příklad

$$b(x) = -\sum_j \frac{1}{g_j(x)} \tag{3}$$

$$b(x) = -\sum_j \log(-g_j(x)) \tag{4}$$

2.4 Penalizačně-bariérové (SUMT)

Rozdělíme množinu {1 ... m } na I_1 disjunktní I_2. Řešíme posloupnost úloh:

$$\min\{f(x) + \alpha_k \sum_{j \in I_1} \varphi(g_j(x)) + \frac{1}{\beta_k} \sum_{j \in I_2} \psi(g_j(x))\} \tag{1}$$

3 Metody hledání lokálního minima

3.1 Gradientní

Podle věty o přírůstku funkce funkce roste nejvíce ve směru gradientu:

$$f(x) = f(x_0) + \nabla f(x_0 + \theta(x - x_0))^T (x - x_0) \tag{1}$$

Počítáme:

$$x_{k+1} = x_k - \zeta \nabla f(x_k), \text{ kde } \zeta \text{ je řešením} \tag{2}$$

$$\min_{\zeta \leq 0} f(x_k - \zeta \nabla f(x_k)) \tag{3}$$

3.2 Newtonova metoda

$$f(x) \doteq f(x_k) + \nabla f(x_k)(x - x_k) + \frac{1}{2}(x - x_k)\nabla^2 f(x_k)(x - x_k) \tag{1}$$
$$\nabla f(x) = 0 \tag{2}$$
$$0 + \nabla f(x_k) + \nabla^2 f(x_k)(x - x_k) - 0 \tag{3}$$
$$x_{k+1} = x_k - \nabla f(x_k)(\nabla^2 f(x_k))^2 \tag{4}$$
$$x_1 = 0 \tag{5}$$

3.3 Lemkeho metoda

Nápad: vezmeme výchozí řešení x a vzdálenost ρ, metodou půlení přibližujeme.
Požadavky: $\{x|f(x) \leq f(x_1)\}$ je omezená.
Problém vhodně zvolit x_1.

Metoda K x_1 sestavíme $x_1 + \rho e^i$ pro $\rho > 0$ a spočítáme funkční hodnoty $f(x_1 + \rho e_i)$ a porovnáme s funkční hodnotou v x_1. Pokud:

1. $f((x_1 + \rho e_i) \geq f(x_1) \quad \forall i \quad \Rightarrow \quad \rho = \frac{\rho}{2}$
2. $\exists j$: $y := f(x_1 + \rho e_j) < f(x_1) \quad \Rightarrow$ opakujeme pro y

4 Lineární programování

Definice Úlohou pro lineární programování v rovnicovém tvaru rozumíme úlohu:

$$\min_M c^T x, \quad M = \{x|Ax = b, x \geq o\}, \tag{1}$$
$$\text{kde} \quad A \in \mathbb{R}^{m\times n}, b \in \mathbb{R}^m, c \in \mathbb{R}^n, \operatorname{rank}(A) = m, 1 \leq m < n, b \geq 0 \tag{2}$$

Definice Úlohou lineárního programování normálním tvaru rozumíme

$$\min_M c^T x, M = \{x|Ax \leq b, x \geq 0\}, \tag{3}$$
$$\text{kde} \quad A \in \mathbb{R}^{m\times n}, b \in \mathbb{R}^m, c \in \mathbb{R}^n, n, m \geq 1 \tag{4}$$

4.1 Podprostor

Tvrzení Množina

$$R = \{x \in \mathbb{R}^n | a^T x = b, a \neq 0\} \tag{1}$$

představuje podprostor dimenze $n - 1$ nazývaný **nadrovinou**.

Důkaz Protože $a \neq 0 \Rightarrow \exists a_1 \neq 0 \Rightarrow x_1 = \frac{b}{a_1} - \sum_{i=1}^{n} \frac{a_i}{a_1} x_i$. Tedy máme $n-1$ LN vektorů v R.

Tvrzení Množina $R^{n-\alpha} = \{x \in \mathbb{R}^n | a_i^T x = b_i, i + 1 ... \alpha\}$ představuje podprostor R^n dimenze $n - \alpha$.

4.2 Poloprostor

Definice Pro libovolnou nadrovinu $R = \{x \in \mathbb{R}^n | a^T x = b, a \neq b\}$ nazýváme:

1. $H^+ = \{x \in \mathbb{R}^n | a^T x > b\}$ otevřeným kladným (pravým) poloprostorem $\mathbb{R}^n$ příslušným R.
2. $H^+ = \{x \in \mathbb{R}^n | a^T x < b\}$ otevřeným záporným (levým) poloprostorem $\mathbb{R}^n$ příslušným R.
3. $\overline{H^+} = \{x \in \mathbb{R}^n | a^T x \geq b\}$ uzavřeným kladným (pravým) poloprostorem $\mathbb{R}^n$ příslušným R.
4. $\overline{H^+} = \{x \in \mathbb{R}^n | a^T x \leq b\}$ uzavřeným záporným (levým) poloprostorem $\mathbb{R}^n$ příslušným R.

Tvrzení Platí:

1. $H^+ \cup H^- \cup R = \mathbb{R}^n$
2. $\overline{H^+} = H^+ \cup R$, $\overline{H^-} = H^- \cup R$
3. $H^+ \cap H^- = H^+ \cap R = H^- \cap R = \emptyset$

Tvrzení Platí:

$$\dim \mathcal{H}_i^+ = \dim \overline{\mathcal{H}_i^+} = n \tag{1}$$
$$\text{kde } \mathcal{H}_i^+ = \{x \in \mathbb{R}^n | x_i > 0\} \tag{2}$$
$$\overline{\mathcal{H}_i^+} = \{x \in \mathbb{R}^n | x_i \geq 0\} \tag{3}$$

Důkaz V každém takovém prostoru leží všechny jednotkové vektory.

Tvrzení Platí:

$$\dim \bigcap_{i=1}^{n} \overline{\mathcal{H}_i^+} = \dim \bigcap_{i=1}^{n} \mathcal{H}_i^+ = n \tag{4}$$

Důsledek Množina přípustných řešení lineárního programování:

$$M = \{x|Ax = b, x \geq 0\} \tag{5}$$

kde $b(A) = m$, $a \leq m < n$.
Taktéž lze zapsat jako:

$$M = \mathbb{R}^n - m \cap \bigcap_{i=1}^{n} \overline{\mathcal{H}_i^+} \tag{6}$$

Definice Každou množinu $M \subset \mathbb{R}^n$, která se deá popsat jako průnik konečného počtu nadrovin a uzavřených poloprostorů nazýváme konvexním polyedrem.