1 Duální simplexová metoda

Autor: Markéta Popelová
Datum: 8.5.2011
Předmět: Základy spojité optimalizace

Zadání

Mějme matici $\mathbf{A} \in \mathbb{R}^{m \times n}$ a primární úlohu lineárního programování v normálním tvaru (P) a k ní příslušnou duální úlohu (D):

$$(P)\ \max_{M_P} \mathbf{c}^T\mathbf{x},\ M_P = \{\mathbf{x} \in \mathbb{R}^n | \mathbf{A}\mathbf{x} \leq \mathbf{b}, \mathbf{x} \geq \mathbf{0}\}$$

$$(D)\ \min_{M_D} \mathbf{b}^T\mathbf{y},\ M_D = \{\mathbf{y} \in \mathbb{R}^m | \mathbf{A}^T\mathbf{y} \geq \mathbf{c}, \mathbf{y} \geq \mathbf{0}\}$$

Ekvivalentní úlohy

Prvně vytvořme úlohy (P') a (D') ekvivalentní k (P) a (D) tak, že přidáme nové proměnné

$$\mathbf{x}' = (x_{n+1}, \dots, x_{n+m})$$

$$\mathbf{y}' = (y_{m+1}, \dots, y_{m+n})$$

a definujeme úlohy jako

$$(P')\ \max_{M_P} \mathbf{c}^T\mathbf{x},\ M'_P = \{(\mathbf{x}, \mathbf{x}') \in \mathbb{R}^{n+m} | \mathbf{A}\mathbf{x} + \mathbf{x}' = \mathbf{b}, \mathbf{x}, \mathbf{x}' \geq \mathbf{0}\}$$

$$(D')\ \min_{M_D} \mathbf{b}^T\mathbf{y},\ M'_D = \{(\mathbf{y}, \mathbf{y}') \in \mathbb{R}^{n+m} | \mathbf{A}^T\mathbf{y} - \mathbf{y}' = \mathbf{c}, \mathbf{y}, \mathbf{y}' \geq \mathbf{0}\}$$

Myšlenka DSM

Budeme procházet přípustná bázická řešení duální úlohy. Jim budou odpovídat nepřípustná bázické řešení primární úlohy se stejnou hodnotou cílové funkce. Tuto hodnotu budeme snižovat, čímž se bude zlepšovat hodnota cílové funkce duálu, avšak zhoršovat hodnota cílové funkce primáru. Jakmile dostaneme přípustné bázické řešení primární úlohy, ukážeme, že to je optimum.

Tabulka duální simplexové metody je redukovaná, tedy neobsahuje jednotkovou podmatici za bázické sloupce. Ve sloupci $\mathbf{d_0}$ najdeme hodnoty primárních bázických proměnných a bude odpovídat kriteriálnímu řádku. Uvědomme si, že to, že je primární bázické řešení nepřípustné, znamená, že má alespoň jednu složku zápornou, tedy $\mathbf{d_0} \ngeq 0$. Tedy jakmile bude $\mathbf{d_0} \geq 0$, budeme v optimu (1. DSM).

V řádku $\mathbf{d}*_\mathbf{0}$ najdeme hodnoty duálních bázických proměnných. Ty mají být po celou dobu přípustné, tedy $\mathbf{d}*_\mathbf{0} \geq 0$. Hodnota d_{00} odpovídá hodnotě cílové funkce primární úlohy s primárními bázickými proměnnými, stejně jako hodnotě cílové funkce duální úlohy s duálními bázickými proměnnými.

Na tabulku tedy můžeme nahlížet jako simplexovou tabulku pro duální úlohu, přičemž je vše transponované (hodnoty jsou v řádku místo ve sloupci, nemáme kriteriální řádek, ale sloupec, atd.).

Algoritmus (Duální simplexová metoda)

1. Inicializace.

2. Transformace.

   (a) Pokud $(\mathbf{d_0} \geq 0) \Rightarrow \mathbf{x} = (\mathbf{x_B}, \mathbf{x_N}) = (\mathbf{d_0}, \mathbf{0})$ je optimum primární úlohy s hodnotou cílové funkce $d_{00} \Rightarrow$ konec.

   (b) Pokud $(\exists i \in B : d_{i0} < 0\ \&\ D_{i*} \geq 0) \Rightarrow$ (P) ani (D) nemají optimální řešení $\Rightarrow$ konec.

(c) Zvolme pivota D_{sr} a proveďmě transformaci $\mathbf{D}$ podle tohoto pivota.

$$s := \min\{i \in B \mid d_{i0} < 0\}$$
$$r := \min\left\{k \in N \;\middle|\; \frac{d^*_{0k}}{|D_{sk}|} = \min\left\{\frac{d^*_{0j}}{|D_{sj}|} \;\middle|\; j \in N, D_{sj} < 0\right\}\right\}$$

1. DSM

Nechť $\mathbf{d_0} \geq 0$. *Pak* $\mathbf{x} = (\mathbf{x_B}, \mathbf{x_N}) = (\mathbf{d_0}, \mathbf{0})$ *je optimální řešení primární úlohy a* $\mathbf{y} = (\mathbf{y_B}, \mathbf{y_N}) = (\mathbf{d}*_\mathbf{0}, \mathbf{0})$ *je optimální řešení duální úlohy se společnou hodnotou cílové funkce* d_{00}.

Důkaz

Algoritmus nám zaručuje, že $\mathbf{y}$ je přípustné bázické řešení[1] a že $\mathbf{d_0}$ jsou hodnoty $\mathbf{x_B}$. Jsou-li všechny kladné, máme navíc přípustné řešení primární úlohy a dle principu duality se jedná o optimální řešení se společnou hodnotou cílové funkce, která je d_{00} dle volby simplexové tabulky. □

2. DSM

Nechť $\exists i \in B : d_{i0} < 0$ & $D_{i*} \geq 0$. *Pak (P) ani (D) nemají optimální řešení.*

Důkaz

Vezměme si libovolné $\mathbf{x} \in M'_P$. Pro něj platí vztah $\mathbf{x_B} = \mathbf{d_0} - \mathbf{D}\mathbf{x_N} \geq \mathbf{0}$[2], kde B jsou indexy aktuálních bázických proměnných, N indexy aktuálních nebázických proměnných. Tedy pro i-tou složku platí:

$$x_i = d_{i0} - D_{*i}\mathbf{x_N} < 0,$$

neboť $d_{i0} < 0$ z předpokladů věty, $D_{i*} \geq 0$ taktéž z předpokladů věty a $\mathbf{x_N} \geq 0$ z definice M'_P. Tedy i-tá složka každého přípustného řešení M'_P je záporná, tedy $M'_P = \emptyset \Rightarrow M_P = \emptyset$. Tedy (P) nemá optimální řešení a dle principu duality ani (D) nemá optimální řešení. □

3. DSM

Nechť nejsou splněny předpoklady 1. ani 2. DSM. Nechť

$$s := \min\{i \in B \mid d_{i0} < 0\}$$
$$r := \min\left\{k \in N \;\middle|\; \frac{d^*_{0k}}{|D_{sk}|} = \min\left\{\frac{d^*_{0j}}{|D_{sj}|} \;\middle|\; j \in N, D_{sj} < 0\right\}\right\}.$$

Pak po jedné transformaci tabulky s pivotem D_{sr} *dostaneme nové bázické řešení* $\mathbf{x}'$ *s bázickými hodnotami* $\mathbf{d}'_\mathbf{0}$ *a nové přípustné bázické řešení* $\mathbf{y}'$ *s hodnotami* $\mathbf{d}'^*_\mathbf{0}$ *a společnou hodnotou cílových funkcí*

$$\mathbf{c}^T\mathbf{x}' = \mathbf{b}^T\mathbf{y}' = d'_{00} \leq d_{00}.$$

[1] Jedná se sice o řešení (D') a (P'), ale to jsou ekvivalentní úlohy k (D) a (P).)

[2] Toto bychom měli dokázat, že to je zachování po celou dobu algoritmu.

Důkaz

Nejdříve ukažme, že $\mathbf{d_0'}$ jsou hodnoty $\mathbf{x_B'}$. Z indukčního předpokladu víme, že

$$\mathbf{d_0} = \mathbf{A_B^{-1} b},$$
$$\mathbf{D} = \mathbf{A_B^{-1} A_N}.$$

Podle stejné úvahy jako u SM platí, že

$$\mathbf{d_0'} = \mathbf{A_{B'}^{-1} b},$$
$$\mathbf{D'} = \mathbf{A_{B'}^{-1} A_{N'}}.$$

Z první rovnosti vidíme, že $\mathbf{d_0'}$ jsou hodnoty nových bázických proměnných $\mathbf{x_B'}$. Dále $\mathbf{d{*}_0}$ udržujeme tak, aby platilo

$$\mathbf{d{*}_0} = (\mathbf{A^T})_{\mathbf{B'}}^{-1}\mathbf{c}.$$

Dále se ukáže i ten zbytek, který v podstatě vyjde z volby pivota. □

Parametrické programování

Obor řešitelnosti jsou taková reálná λ, že s nimi má úloha řešení. Obor stability řešení x_1 jsou taková reálná λ, že s nimi má úloha optimální řešení x_1. Funkce řešitelnosti dané úlohy je $\phi(\lambda) = \min_M (c + \lambda c')^T x$ kde λ je z oboru řešitelnosti. Dá se dokázat, že pro jedno x_1 existuje uzavřený maximální obor stability. Dále že pro λ mimo obor řešitelnosti existuje otevřený interval okolo, který také spadá mimo obor řešitelnosti. A nakonec, že pro jedno x_1 s nějakým oborem stability existuje sousední x_2 s navazujícím oborem stability.

Úlohy na volný extrém

Prostě taková analýznická úloha, kdy se hledá globální extrém funkce na $\mathbb{R}^n$. Hledají se stacionární body (nulová derivace, resp. nulový gradient (vektor parciálních derivací)). Pak se počítá Hessián a ověřuje se definitnost ve stacionárních bodech. Jsou i nějaké numerické metody.

Nelineární programování

Konvexní programování – lokální minimum je i globální. Pro explicitně-kvazikonvexní to platí taktéž, ale pro konvexní ne. Konvexní $\Rightarrow$ explicitně-kvazikonvexní $\Rightarrow$ konvexní Pseudokonvexní: pomocí gradientu f v pseudokonvexním bodě lze určit absolutní minimum f na M.

1.0.1 Metody

1. Metody přípustných směrů, např. metoda Franka a Wolfa. Fajn pro konvexní M, aby šlo rychle určit, zda jsi uvnitř. Cílová fce f libovolná, ale $f \in \mathcal{C}^1$.

2. Metody vnější aproximace, jako sečné nadroviny. Najde se konvexní polyedr, co obsahuje M, najde se jeho optimum. A když tam neleží, tak se sestrojí sečná nadrovina a uřízne se kus konvexního polyedru. Fajn pro jednoduchou célovou funkci, ideálně linerání, páč se pak lépe hledá optimum na tom konvexním polyedru.

3. Metody vnitřní aproximace. Cílová funkce se aproximuje nějakou snazší.

4. Metody založené na K.-T. podmínkách. Vhodné pro kvadratické programování, páč z toho pak vznikne soustava rovnic a nerovnic.

5. Branch and Bound. Dělí velkou úlohy na posloupnost menších. Fajn pro konečnou M.

6. Penalizační metody. Vyjde se z libovolného bodu a pak se přibližuje k M. Cílová funkce je nějak min f + $\alpha_k p(\mathbf{x})$ a jelikož $\alpha_k \to \infty$, tak potřebujeme, aby $p \to \infty$, což znamená, že se přibližujeme k M.

7. Bari=rové metody. Musí se najít vnitřní bod M. Pak se nějak zlepšuje hodnota cílové funkce. Výhodné je, že lze skončit a mít přijatelné řešení.

1.0.2 Metoda Franka a Wolfa

f je libovolná konvexní z $\mathcal{C}^1$ funkce, M je omezený konvexní polyedr (tedy dán soustavou nerovnic).

1. Najdeme výchozí bod $x^1 \in M$. Když neexistuje, tak hotovo (M prázdná).

2. Sestavíme tečnou nadrovinu M v x^1 jako $\nabla f(x^1)(x - x^1)$, což je lineární funkce. Stačí uvažovat $\nabla f(x^1)^T x$, páč minimum nezávisí na konstantě. Najdeme minimum na M s touto cílovou lineární funkcí pomocí SM, označíme ho $\hat{x}$. Pokud x^1 nebylo optimum, tak máme přípustný směr $\hat{x} - x^1$, který alespoň kousek klesá, tedy derivace f v tomto směru z bodu x^1 je $\nabla f(x^1)^T (x - x^1) \leq 0$. (Kdyby byla rovna nule, jsme v optimu. Pokud $\nabla f(\hat{x})^T (x - x^1) \leq 0$, určíme $x^2 := \hat{x}$ a pokračujeme krokem 1. Jinak hledáme min $f(x^1 a \lambda(\hat{x} - x^1))$ pro $\lambda \in \langle 0, 1, \rangle$, což je funkce jedné proměnné na kompaktní množině, tedy má dle Weierstrasse minimum a lze ho jasně určit hnusným vzorcem s derivacemi. Když ho tam dosadíme, získáme x^2 pro další krok.

Vícekriteriální programování

Cílovou funkci tvoří vektor s funkcí $f_i : \mathbb{R}^n \to \mathbb{R}$. Máme 4 možnosti, jak hodnotit, které řešení je nejlepší:

1. Ideální řešení. Je maximem pro všechny f_i. To typicky nenastává.

2. Dominantní řešení. Vybereme si jednu f_i a na ostatní se vybodnem.

3. Kompromisní řešení. Uživatel nám ta f_i nějak ohodnotí (třeba jim dá váhy), tzn. dá nám funkci $F(f_i)$ a my pak řešíme maximum této funkce. Dělí se to ještě podle toho, kdy tyto preference dostaneme:
   (a) Nikdy. Nějak je tedy vymyslíme sami, v podstatě spíše z matematického hlediska.
   (b) Před výpočtem. Fajn, rozvnou víme, co řešit.
   (c) Při výpočtu. To je pak prý nějak složité.
   (d) Po skončení výpočtu. Na začátku si je nějak zvolíme a po konci to zopakujeme s preferencemi uživatele.

4. Eficientní řešení. Neexistuje jiné takové, že by bylo ve všech f_i alespoň stejně dobré a v alespoň jedné f_j by bylo lepší.

Eficientní řešení

Podstatný je skalární parametrický ekvivalent obecné úlohy VP. Platí totiž, že $\bigcup_{\lambda > \mathbf{0}} M_{opt}(\lambda)$, tedy sjednocení přes všechna λ množin všech optimálních řešení pro dané λ, patří celá do množiny všech eficientních řešení ϵ. Důkaz je snadný, důsledky jsou důležité. Dokonce pro lineární VP se to rovná, pro konvexní sice ne, ale lze ϵ odhadnout zhora sjednocením stejné množiny přes $\lambda \geq \mathbf{0}$.

Pro tento typ řešení existují algoritmy dialogu. Ten první si zvolí nějaká vhodná λ_1, λ_2, řeší pak maximum $\lambda_1 + t(\lambda_2 - \lambda_1)^T \mathbf{f}(\mathbf{x})$, kde t je volitelný parametr. Pro $t = 0$ to vyřeší a nabídne uživateli optimální řešení společně s několika dalšími informacemi. Uživatel se může rozhodnout, zda:

1. pokračovat v tomto směru, tedy zvětšit t,

2. pokračovat v jiném směru, tedy změnit λ_2,

3. vrátit se k dříve dosaženému optimu a příslušnému λ,

4. změnit λ_1, λ_2,

5. skončit.

Druhý algoritmus dialogu se použije tehdy, co je uživatel dlouho nespokojen. Nějak se tam zavede ještě nějaké K a když je z nějakého hlesika (jedné f_i) výsledek již dobrý, tak se dané f_i už nesnažíme zlepšit, ale snažíme se ho příliš nezhoršit.

Kompromisní řešení

Když nám uživatel nic nedá, použije se metoda globální cílové funkce. Pro každé i se najde vyřeší maximalizace $f_i(x)$. A pak hledá řešení, které se minimálně odchýlí od každé z těchto hodnot. Dá se ukázat, že toto je eficientní řešení. Pak jsou ještě nějaké další možnosti.

Dynamické programování

Systém je to, co zkoumáme v časovém intervalu $\langle t_1, t_2 \rangle$. *Stav* systému v čase t je popsaný n-ticí *stavových proměnných.* $X \subset \mathbb{R}^n$ je množina přípustných stavů. Stavy se mění na základě *rozhodnutí* u, což je m-tice *rezhodovacích proměnných.* $U \subset \mathbb{R}^m$ je množina přípustných rozhodnutí.

Rozhodování mohou nastávat buď v libovolném čase (čas je spojitý), což vede na Pontrjaginův princip maxima, nebo jen v diskrétních časových okamžicích, což budeme uvažovat v další části. Dá se pak totiž ukázat, že platí Ballmanův princip optimality, který říká, že každá podstategie optimální strategie je optimální.

Def. Chová-li se systém v časovém intervale $\langle t_1, t_2 \rangle$ dle následujících podmínek, říkáme, že v $\langle t_1, t_2 \rangle$ probíhá N-stupňový diskrétní proces se stavovou transformací $\forall i \in \langle t_1, t_2 \rangle x_{i+1} = T_i(x_i, u_i)$. Tím je myšleno, že stav systému v čase i je dán nějakou funkcí předchozího stavu a předchozího rozhodnutí. A teď ty podmínky:

1. Interval $\langle t_1, t_2 \rangle$ je rozdělen pomocí $t_1, \ldots, t_N$ časových okamžiků, kde $N > 0, N \in \mathbb{N}$.

2. Stav systému se mění jen v t_i, kde $i \in \langle 1, N \rangle \cap \mathbb{Z}$.

3. $\forall i$ v intervalu $\langle t_{i-1}, t_i)$ máme stav x_i, před prvním rozhodnutím u_1 máme počáteční stav x_1 a poslední stav je x_{n+1}.

4. Stav x_{i+1} závisí jen na x_i, u_i. Píšeme $\forall i \in \langle t_1, t_2 \rangle x_{i+1} = T_i(x_i, u_i)$.

Čas máme rozdělen diskrétně na intervaly $t_1, \ldots, t_N$. V každém tomto intervalu se změní stav na základě rozhodnutí u, což je m-tice nějakých reálných čísel. Přípustná strategie je posloupnost $u_1, \ldots u_N$, že vede z přípustného stavu do přípustného stavu po přípustných stavech přes příslušná rozhodnutí. Zajímá nás především úloha, kdy máme dán vstupní stav x_1 a máme za N kroků nasbírat co nejvíce bodů přechody mezi stavy, přičemž účelová funkce závisí vždy na stavu, odkud jsme přišli, a rozhodnutí, které nás k tomu vedlo, a čase, ve který se to děje. Dynamické programování využívá Bellmanův princip optimality, že každá podstategie optimální strategie je optimální.

1. Pro každý předposlední stav vybereme poslední stav, které spojuje hrana nejvyšší hodnoty.

2. Pro každý předpředposlední stav vybereme nejlepší předposlední stav, tedy maximalizujeme hodnotu hrany do tohoto stavu + hodnotu cílové funkce v tamtom stavu, kterou už máme spočítanou.

3. …atd. až získáme optimální hodnotu pro první vrchol.

Teorie her

Máme matici A, řádkový hráč si vybírá i index řádku, sloupcový j index sloupce, řádkovému se přičte a_{ij}, což může být i záporné. Strategie hráče je vektor pravděpodobností výběru daného indexu. Jsou-li dané strategie p, q, je očekávaná hodnota $E(p,q) = \sum_{i,j=1}^{n} p_i a_{ij} q_j$.

Věta říká, že existují optimální strategie p^*, q^*, že $E(p^*, q) \geq E(p^*, q^*) \geq E(p, q^*) = v$. Tedy se jim nevyplatí hrát jinak než tou optimální strategií. Tedy vždy existuje nějaké v, které může hráč uhrát, ať bude hrát ten druhý jakkoliv.

Nás zajímá optimální strategie p^*, q^*. Nějak ukážeme, že stačí uvažovat jednotkové vektory. Dále z toho sestavíme úlohu lineárního programování, kde p^* bude optimální řešení primáru a q^* bude optimální řešení duálu.