Lineární modely – 10. přednáška

Lineární programování

Ondřej Klíma

25. 4. 2013

Osnova přednášky

- Simplexy, rovnoběžnostěny a další konvexní množiny
- Úloha lineárního programování
- Dvoudimenzionální příklad
- Simplexový algoritmus

Simplexy

Definice

Nechť $A_0, \ldots, A_k$ je $k+1$ bodů afinního prostoru $\mathcal{A}$ v obecné poloze. Pak k-rozměrný simplex $\Delta = \Delta(A_0, \ldots, A_k)$ generovaný těmito body je definován jako množina všech afinních kombinací bodů A_i s pouze nezápornými koeficienty, tzn.

$$\Delta = \{t_0 A_0 + t_1 A_1 + \cdots + t_k A_k \mid t_i \in [0,1], \sum_{i=0}^{k} t_i = 1\}.$$

Jednorozměrný simplex je úsečka, dvourozměrný trojúhelník.

Nestačí nám to. Chceme čtyřúhelníky a složitější prostorová tělesa.

Rovnoběžnostěn

Definice

Nechť $u_1, \dots, u_k$ jsou libovolné vektory v zaměření $\mathbb{R}^n$, $A \in \mathcal{A}_n$ je libovolný bod. Rovnoběžnostěn $\mathcal{P}_k(A; u_1, \dots, u_k) \subseteq \mathcal{A}_n$ je množina

$$\mathcal{P}_k(A; u_1, \dots, u_k) =$$

$$= \{A + c_1 u_1 + \dots + c_k u_k \mid 0 \leq c_i \leq 1, i = 1, \dots, k\}.$$

Jsou-li vektory $u_1, \dots, u_k$ nezávislé, hovoříme o k-rozměrném rovnoběžnostěnu $\mathcal{P}_k(A; u_1, \dots, u_k) \subseteq \mathcal{A}_n$.

Tj. rovnoběžník v $\mathbb{R}^3$ je rovnoběžnostěn (dimenze 2).

Konvexní množiny

Definice

Podmnožina M afinního prostoru se nazývá konvexní, jestliže s každými svými dvěma body A, B obsahuje i celou úsečku $\Delta(A, B)$.

- Každá konvexní množina obsahuje s každými $k + 1$ body v obecné poloze i celý jimi definovaný simplex.
- Konvexními množinami jsou např.:
  - prázdná podmnožina,
  - afinní podprostory,
  - úsečky, polopřímky $p = \{P + t \cdot v;\ t \geq 0\}$,
  - obecněji k-rozměrné poloprostory $\alpha = \{P + t_1 \cdot v_1 + \cdots + t_k \cdot v_k \mid t_1, \ldots, t_k \in \mathbb{R}, t_k \geq 0\}$,
  - úhly v dvojrozměrných podprostorech $\beta = \{P + t_1 \cdot v_1 + t_2 \cdot v_2 \mid t_1 \geq 0, t_2 \geq 0\}$,
  - další: nekonečný jehlan, kužel, koule, mnohostěny.

Konvexní obal

Přímo z definice také plyne, že průnik libovolného systému konvexních množin je opět konvexní množina.

Definice

Průnik všech konvexních množin obsahujících danou množinu M nazýváme konvexní obal $\mathcal{K}(M)$ množiny M.

Věta

Konvexní obal libovolné podmnožiny $M \subseteq \mathcal{A}$ je

$$\mathcal{K}(M) = \{t_1 A_1 + \cdots + t_s A_s \mid s \in \mathbb{N}, \sum_{i=1}^{s} t_i = 1,\ t_i \geq 0, A_i \in M\}$$

Konvexní obaly konečných množin bodů se nazývají konvexní mnohostěny. Jsou-li definující body $A_0, \ldots, A_k$ konvexního mnohostěnu v obecné poloze, dostáváme právě k-rozměrný simplex.

Konvexní obal – příklad

Příklad

Rozhodněte, zda leží bod $X = [2, 1, 0]$ uvnitř konvexního obalu bodů $[0, 2, 1]$, $[1, 0, 1]$, $[3, -2, -1]$, $[-1, 0, 1]$.

Sestavíme nehomogenní lin. soustavu, pro koeficienty t_1, t_2, t_3, t_4, afinní kombinace daných bodů, která dává bod X (jsou určeny jednoznačně, pokud dané body neleží v rovině).

$$\begin{pmatrix} 0 & 1 & 3 & -1 \\ 2 & 0 & -2 & 0 \\ 1 & 1 & -1 & 1 \\ 1 & 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} t_1 \\ t_2 \\ t_3 \\ t_4 \end{pmatrix} = \begin{pmatrix} 2 \\ 1 \\ 0 \\ 1 \end{pmatrix}.$$

Poslední rovnice udává, že jde o afinní kombinaci. Řešením soustavy dostáváme $(t_1, t_2, t_3, t_4) = (1, 0, 1/2, -1/2)$, nejedná se tedy o konvexní kombinaci. Bod X neleží v konvexním obalu.

Lineární optimalizační úlohy

Příklad (Modifikace příkladu 3.1 ze skript)

Firma vyrábí šroubky a matice.

- Šroubky i matice jsou lisovány – vylisování krabičky šroubků trvá 1 minutu, krabička matic je lisována 2 minuty.
- Šroubky i matice balí do krabiček – krabička šroubků se balí 1 minutu, krabička matic 4 minuty.
- Firma má k dispozici 2 hodiny času pro lisování a 3 hodiny času pro balení výrobků.
- Vzhledem k poptávce chce firma vyrobit maximálně o 30 krabiček matic více než krabiček šroubků.
- Firma nemůže vyrobit více než 110 krabiček šroubků.
- Zisk z jedné krabičky je 20 Kč (šroubky) a 100 Kč (matice).

Kolik krabiček šroubků a matic má firma vyrobit, chce-li dosáhnout maximálního zisku?

Matematická formulace příkladu

- Buď x_1 počet krabiček šroubků, x_2 počet krabiček matic.
- Ze zadání dostáváme omezující podmínky:

$$\begin{aligned} x_1 + 2x_2 &\leq 120 \\ x_1 + 4x_2 &\leq 180 \\ x_2 &\leq 30 + x_1 \\ x_1 &\leq 110 \end{aligned}$$

- Účelová funkce (funkce udávající zisk při daném počtu vyrobených šroubků a matic) je $20x_1 + 100x_2$.
- Předchozí soustava nerovnic zadává v $\mathbb{R}^2$ určitou oblast (implicitně $x_1 \geq 0$ a $x_2 \geq 0$) a optimalizace zisku znamená najít v této oblasti bod (případně body) s nejvyšší hodnotou účelová funkce. Tj. lze řešit „geometricky".
- Optimální řešení: průsečík přímek $p : x_1 + 4x_2 = 180$ a $q : x_2 = 30 + x_1$. Tj. bod $x_1 = 12, x_2 = 42$.

Formulace obecné úlohy lineárního programování

Definice

Obecnou úlohou lineárního programování rozumíme

$$\max\{c \cdot x^T \mid A \cdot x^T \leq b^T\}$$

kde $c = (c_1, \ldots, c_m)$ je vektor účelové funkce, $x = (x_1, \ldots, x_m)$ vektor promněnných, $A \in Mat_{k,m}(\mathbb{R})$ matice soustavy lineárních omezení a $b \in \mathbb{R}^k$ vektor (omezení).

V našem příkladě: $m = 2$, $c = (20, 100)$a $A \cdot x^T \leq b^T$ je tvaru

$$\begin{pmatrix} 1 & 2 \\ 1 & 4 \\ -1 & 1 \\ 1 & 0 \\ -1 & 0 \\ 0 & -1 \end{pmatrix} \cdot \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \leq \begin{pmatrix} 120 \\ 180 \\ 30 \\ 110 \\ 0 \\ 0 \end{pmatrix}$$

Manipulace s úlohami lineárního programování

- Minimalizaci nemá smysl formulovat zvlášť, neboť $\min\{c \cdot x^T \mid \dots\}$ je totéž co $\max\{-c \cdot x^T \mid \dots\}$.
- Lze uvažovat i rovnice $a \cdot x^T = p$, které jsou ekvivalentní s dvojicí nerovnic $a \cdot x^T \leq p$, $-a \cdot x^T \leq -p$.
- Naopak libovolnou nerovnici lze zavedením nové proměnné převést na rovnici s dodatečnou podmínkou

$$(a_1, \dots, a_m) \cdot (x_1, \dots, x_m)^T \leq b_0 \iff$$
$$(a_1, \dots a_m, 1) \cdot (x_1, \dots, x_m, y)^T = b_0, y \geq 0\,.$$

- Požadavek nezápornosti řešení lze rozšířit na všechny proměnné, neboť místo proměnné x lze vzít $x = x' - x''$, kde $x', x'' \geq 0$.

Definice

Standardní úlohou lineárního programování rozumíme

$$\max\{c \cdot x^T \mid A \cdot x^T = b^T, x \geq 0\}\,.$$

Příklad ve tvaru standardní úlohy

Máme 4 nerovnosti, takže potřebujeme 4 nové proměnné: y_1, y_2, y_3, y_4. Pak $c = (20, 100, 0, 0, 0, 0)$ a $A \cdot x^T = b^T$ je tvaru

$$\begin{pmatrix} 1 & 2 & 1 & 0 & 0 & 0 \\ 1 & 4 & 0 & 1 & 0 & 0 \\ -1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} x_1 \\ x_2 \\ y_1 \\ y_2 \\ y_3 \\ y_4 \end{pmatrix} = \begin{pmatrix} 120 \\ 180 \\ 30 \\ 110 \end{pmatrix}$$

Tabulkový zápis standardní úlohy:

$$\begin{array}{rrrrrr|r} -20 & -100 & 0 & 0 & 0 & 0 & 0 \\ \hline 1 & 2 & 1 & 0 & 0 & 0 & 120 \\ 1 & 4 & 0 & 1 & 0 & 0 & 180 \\ -1 & 1 & 0 & 0 & 1 & 0 & 30 \\ 1 & 0 & 0 & 0 & 0 & 1 & 110 \end{array}$$

Existence řešení úlohy lineárního programování

Všimněme si, že body splňující systém omezení tvoří konvexní množinu. Jde totiž o průnik poloprostorů.

Pro každou úlohu lineárního programování nastane právě jedna z následujících tří možností:

- Optimální řešení neexistuje, protože neexistuje žádné přípustné řešení.
- Optimální řešení neexistuje z důvodu neohraničenosti. (Úloha má ale přípustná řešení).
- Existují optimální řešení (může jich být více).

My se zpravidla budeme potýkat se situacemi třetího typu, protože jednak bývají přirozeně „omezeny zdroje“ a navíc je k dispozici „triviální“ (nulové) přípustné řešení.

Příklady z praxe na úlohy lineárního programování

V různých ekonomických modelech chceme minimalizovat náklady nebo maximalizovat zisk při omezeních.

- *Úlohy finančního plánování*, související s optimalizací portfolia. Určujeme přitom objemy investic do jednotlivých investičních variant s cílem držet se daných omezení na rizika a optimalizovat přitom zisk, resp. při očekávaném objemu minimalizovat rizika.
- *Marketingové aplikace*, např. alokace nákladů na reklamy v různých médiích nebo umísťování reklam do časových termínů. Omezujícími podmínkami bude disponibilní rozpočet, rozložení cílových skupin apod.
- *Výživové problémy*, tj. návrh dávek různých komponent výživy s daným složením a omezujícími požadavky na celkové objemy výživových látek.
- *Personální úlohy*, kdy jsou pracovníci s různými kvalifikacemi a dalšími předpoklady rozdělováni do směn.
- *Distribuce zboží* ze skladů je třeba distribuovat zboží mezi zákazníky, při minimalizaci nákladů na dopravu.

Simplexový algoritmus pro standardní úlohu

- Idea – postupujeme z vrcholu do vrcholu po hranách a vylepšujeme hodnotu účelové funkce.
- Řešíme zjednodušenou úlohu, kde $Ax \leq b$, s $b \geq 0$, tudíž máme „bezpracně“ přípustné nulové řešení.
- Obecně se musí startovní řešení najít, resp. ukázat, že neexistuje. My nebudeme dělat.

Simplexový algoritmus – tabulkový zápis

Tabulkový zápis standardní úlohy:

$$
\begin{array}{rrrrrr|r}
-20 & -100 & 0 & 0 & 0 & 0 & 0 \\
\hline
1 & 2 & 1 & 0 & 0 & 0 & 120 \\
1 & 4 & 0 & 1 & 0 & 0 & 180 \\
-1 & 1 & 0 & 0 & 1 & 0 & 30 \\
1 & 0 & 0 & 0 & 0 & 1 & 110
\end{array}
$$

- Sloupce s pivotem – v pravém sloupci v řádku s pivotem se nachází aktuální hodnota příslušné proměnné. Zde $y_1 = 120$, $y_2 = 180$, $y_3 = 30$, $y_4 = 110$.
- Sloupce bez pivota – příslušná proměnná je nulová. Zde $x_1 = 0$, $x_2 = 0$.
- Aktuální hodnota účelové funkce – pravý horní roh. Zde 0.

Simplexový algoritmus – jeden krok výpočtu

$$\begin{array}{ccc|c}
-c_1 & \dots & -c_n & 0 \\
\hline
a_{11} & \dots & a_{1m} & b_1 \\
\vdots & & \vdots & \vdots \\
a_{k1} & \dots & a_{km} & b_k
\end{array}$$

1. Vybereme první sloupec zleva, který má v záhlaví záporný prvek. Nechť je to j-tý sloupec.
2. V j-tém sloupci vybereme z kladných čísel a_{ij} to, pro které je poměr $\frac{b_i}{a_{ij}}$ minimální.
3. Eliminujeme celý j-tý sloupec této tabulky s pivotem a_{ij} (tzn. elementárními řádkovými transformacemi dané tabulky dosáhneme toho, že ve vybraném sloupci bude číslo 1 na místě a_{ij} a jinak samé nuly).

Poznamenejme, že pravý sloupec zůstává během výpočtu nezáporný.

Simplexový algoritmus – příklad – první krok

$$
\begin{array}{rrrrrr|r}
-20 & -100 & 0 & 0 & 0 & 0 & 0 \\
\hline
1 & 2 & 1 & 0 & 0 & 0 & 120 \\
1 & 4 & 0 & 1 & 0 & 0 & 180 \\
-1 & 1 & 0 & 0 & 1 & 0 & 30 \\
1 & 0 & 0 & 0 & 0 & 1 & 110
\end{array}
\qquad
\begin{array}{r}
\\
120 \\
180 \\
\\
110
\end{array}
$$

$$
\begin{array}{rrrrrr|r}
0 & -100 & 0 & 0 & 0 & 20 & 2200 \\
\hline
0 & 2 & 1 & 0 & 0 & -1 & 10 \\
0 & 4 & 0 & 1 & 0 & -1 & 70 \\
0 & 1 & 0 & 0 & 1 & 1 & 140 \\
1 & 0 & 0 & 0 & 0 & 1 & 110
\end{array}
$$

$x_1 = 110, x_2 = 0, y_1 = 10, y_2 = 70, y_3 = 140, y_4 = 0$

Simplexový algoritmus – příklad – druhý krok

$$\begin{array}{cccccc|c|c}
0 & -100 & 0 & 0 & 0 & 20 & 2200 & \\
\hline
0 & 2 & 1 & 0 & 0 & -1 & 10 & 10 \\
0 & 4 & 0 & 1 & 0 & -1 & 70 & 17,5 \\
0 & 1 & 0 & 0 & 1 & 1 & 140 & 140 \\
1 & 0 & 0 & 0 & 0 & 1 & 110 &
\end{array}$$

$$\begin{array}{cccccc|c}
0 & 0 & 50 & 0 & 0 & -30 & 2700 \\
\hline
0 & 1 & \frac{1}{2} & 0 & 0 & -\frac{1}{2} & 5 \\
0 & 0 & -2 & 1 & 0 & 1 & 50 \\
0 & 0 & -\frac{1}{2} & 0 & 1 & \frac{3}{2} & 135 \\
1 & 0 & 0 & 0 & 0 & 1 & 110
\end{array}$$

$x_1 = 110, x_2 = 5, y_1 = 0, y_2 = 50, y_3 = 135, y_4 = 0$

Simplexový algoritmus – příklad – třetí krok

$$
\begin{array}{cccccc|cc}
0 & 0 & 50 & 0 & 0 & -30 & 2700 & \\
\hline
0 & 1 & \frac{1}{2} & 0 & 0 & -\frac{1}{2} & 5 & \\
0 & 0 & -2 & 1 & 0 & 1 & 50 & 50 \\
0 & 0 & -\frac{1}{2} & 0 & 1 & \frac{3}{2} & 135 & 90 \\
1 & 0 & 0 & 0 & 0 & 1 & 110 & 110
\end{array}
$$

$$
\begin{array}{cccccc|c}
0 & 0 & -10 & 30 & 0 & 0 & 4200 \\
\hline
0 & 1 & -\frac{1}{2} & \frac{1}{2} & 0 & 0 & 30 \\
0 & 0 & -2 & 1 & 0 & 1 & 50 \\
0 & 0 & \frac{5}{2} & -\frac{3}{2} & 1 & 0 & 60 \\
1 & 0 & 2 & -1 & 0 & 0 & 60
\end{array}
$$

$x_1 = 60, x_2 = 30, y_1 = 0, y_2 = 0, y_3 = 110, y_4 = 50$

Simplexový algoritmus – příklad – čtvrtý krok

$$
\begin{array}{cccccc|c c}
0 & 0 & -10 & 30 & 0 & 0 & 4200 & \\
\hline
0 & 1 & -\frac{1}{2} & \frac{1}{2} & 0 & 0 & 30 & \\
0 & 0 & -2 & 1 & 0 & 1 & 50 & \\
0 & 0 & \frac{5}{2} & -\frac{3}{2} & 1 & 0 & 60 & 24 \\
1 & 0 & 2 & -1 & 0 & 0 & 60 & 30
\end{array}
$$

$$
\begin{array}{cccccc|c}
0 & 0 & 0 & 24 & 4 & 0 & 4440 \\
\hline
0 & 1 & 0 & \frac{1}{5} & \frac{1}{5} & 0 & 42 \\
0 & 0 & 0 & -\frac{1}{5} & \frac{4}{5} & 1 & 98 \\
0 & 0 & 1 & -\frac{3}{5} & \frac{2}{5} & 0 & 24 \\
1 & 0 & 0 & \frac{1}{5} & -\frac{4}{5} & 0 & 12
\end{array}
$$

$x_1 = 12, x_2 = 42, y_1 = 24, y_2 = 0, y_3 = 0, y_4 = 98$

Simplexový algoritmus – konec

Konec výpočtu:

- Celé záhlaví je nezáporné, máme optimální řešení.
- Pokud během výpočtu narazíme na sloupec se záporným číslem v záhlaví, který je celý nekladný, pak optimální řešení neexistuje z důvodu neohraničenosti.

V našem příkladě máme optimální řešení $x_1 = 12, x_2 = 42$ s optimální hodnotou účelové funkce 4.440.

Požadavky

- Konvexní obal – zda zadaný bod je vnitřní bod.
- Geometrické řešení úlohy lineárního programování v rovině.
- Algoritmické řešení zjednodušené úlohy lineárního programování. (Jednoduché zadání, kde kromě doplňkových proměnných jsou maximálně 2 původní proměnné.)